SOFTWARE OT&E GUIDELINES

VOLUME I

SOFTWARE TEST MANAGER'S HANDBOOK

FEBRUARY 1981

AIR FORCE TEST AND EVALUATION CENTER
KIRTLAND AIR FORCE BASE
NEW MEXICO 87117

AFTECP-800-1

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFTECP-800-1	2. GOVT ACCESSION NO. AD-A104	3. RECIPIENT'S CATALOG NUMBER 322
4. TITLE (and Subtitle) SOFTWARE OT&E GUIDELINES, Volume I (of five), Software Test Manager's Handbook		5. TYPE OF REPORT & PERIOD COVERED Final, Feb 1981
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) N/A		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS HQ Air Force Test and Evaluation Center AFTEC/TEBC Kirtland AFB NM 87117		10. PROGRAM ELEMENT PROJECT TASK AREA & WORK UNIT NUMBERS N/A
11. CONTROLLING OFFICE NAME AND ADDRESS Same as 9.		12. REPORT DATE Feb 81 13. NUMBER OF PAGES 138
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

AFTEC/TEBC point of contact:
Lt Col Michael A. Blackledge, USAF

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Software
Software Testing
Software Evaluation
Quality Factors
Software Maintenance
Operational Test & Evaluation (OT&E)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Software OT&E Guidelines is a set of handbooks prepared by the Computer/Support Systems Division of the Test and Evaluation Directorate of the Air Force Test and Evaluation Center (AFTEC) for use in the operational test and evaluation (OT&E) of software. Volumes in the set include: 1. Software Test Managers Handbook (AFTEP 800-1); 2. Handbook for the Deputy for Software Evaluation (AFTECP 800-2); 3. Software Maintainability Evaluator's Handbook (AFTECP 800-3); 4. Software Operator-Machine Interface Evaluator's Handbook (cont'd.)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Blk 20 cont'd.

(AFTECP 800-4); 5. Software Support Facility Evaluation Tools User's Handbook (AFTECP 800-5). The Software Test Manager's Handbook was prepared as a guide for the HQ AFTEC software test managers in performing their jobs. It provides a description of the AFTEC approach to OT&E of software. The document is not directive, but rather a reference document for planning OT&E. It contains general instructions and information on the use of various software evaluation tools available to the software test manager, and lessons learned from efforts on earlier programs.

SOFTWARE OT&E GUIDELINES

VOLUME I

SOFTWARE TEST MANAGER'S HANDBOOK

FEBRUARY 1981

AIR FORCE TEST AND EVALUATION CENTER
KIRTLAND AFB, NEW MEXICO 87117

FOREWORD

This volume is one of a set of handbooks prepared by the Computer/Support Systems Division of the Test and Evaluation Directorate, Air Force Test and Evaluation Center (AFTEC) for use in the operational test and evaluation of software. Comments should be directed to AFTEC/TEB, Kirtland AFB, NM 87117. Volumes in the set include:

I. Software Test Manager's Handbook (AFTECP 800-1).

II. Handbook for the Deputy for Software Evaluation (AFTECP 800-2).

III. Software Maintainability Evaluator's Handbook (AFTECP 800-3).

IV. Software Operator-Machine Interface Evaluator's Handbook (AFTECP 800-4).

V. Software Support Facility Evaluation Tools User's Handbook (AFTECP 800-5).

CONTENTS

CONTENTS (continued)

ILLUSTRATIONS

SECTION I

GENERAL INFORMATION FOR SOFTWARE TEST MANAGERS

A. INTRODUCTION.

1. Purpose.

This handbook was prepared as a guide for the HQ AFTEC software test managers in performing their job. It documents numerous activities and bits of information "learned the hard way" but not necessarily passed on to all succeeding software test managers. HQ AFTEC software test managers should not view this document as a directive, but rather as a source of information about OT&E of software and as a reference document to be used in planning for future OT&E. Although this handbook is primarily for HQ AFTEC/ TEBC personnel, individuals from other organizations will find in it a description of the AFTEC approach to OT&E of software. Therefore, it should promote better understanding between AFTEC personnel and individuals from the organizations with which we interface.

AFTEC's approach to OT&E has followed an evolutionary process since 1976. Although there have been some false starts the approach has been improved, becoming more structured and consistent. The evolution will certainly continue.

2. Content of Handbook.

This handbook is divided into three sections.

Section I provides general information on OT&E, AFTEC organization, and the OT&E process--all with a focus on software evaluation and the software test manager.

Section II contains general instructions and information on the use of various software evaluation tools available to the software test

manager, including the software maintainability evaluation questionnaire, the software operator-machine interface questionnaire, and the event trace monitor. Along with the general instructions, references are given for more detailed information.

Section III contains lessons learned from the efforts of software test managers on earlier programs. This part of the handbook is expected to grow as AFTEC gains OT&E experience on more and more software intensive systems.

3. Related Documents.

There are numerous documents that relate to the duties of the software test manager. No attempt has been made in this handbook to extract appropriate portions of all existing regulations, manuals, and operating instructions that apply to software evaluation, the acquisition process, test and evaluation, etc. Occasional reference is made to such documents, but it is advisable for the software test manager to read and be familiar with a number of these directives. The following are of primary importance.

DOD Directive 5000.3	Test and Evaluation
AFR 80-14	Test and Evaluation
AFR 800-14 Vol I	Management of Computer Resources in Systems
AFR 800-14 Vol II	Acquisition and Support Procedures for Computer Resources in Systems
AFM 55-43 Vol I-II	Management of Operational Test and Evaluation
AFTECR 55-1	AFTEC Operations Regulation

In addition, program documentation for the specific system to undergo OT&E will be of interest to the software test manager. Included in this are:

Program Management Directive (PMD)
Program Management Plan (PMP)
Operation and Support Concepts
*Computer Resource Integrated Support Plan (CRISP)
(Ref AFR 800-14)
*Test and Evaluation Master Plan (TEMP)
(Ref DODD 5000.3 and AFR 80-14)

*Operational/Support Configuration Management Procedures (O/S CMP) (Ref AFR 800-14)
*These documents will normally be prepared/revised during full scale engineering development, and the software test manager should have some involvement in the preparation or revision

B. OT&E OF SOFTWARE.

There is wide misunderstanding of exactly what OT&E of software is and what it is not. At one end of the spectrum are those who take the position that there is no such thing as OT&E of software. At the other extreme are those who feel that OT&E of software is a separate and distinct action that can be completely isolated and performed totally by a separate group of specialists. The AFTEC position is that software, when present, is an integral part of the overall system and must be evaluated in that context, yet it requires special emphasis because of the unique nature of software and the difficulty of uncovering software problems. This position is consistent with the basic nature of OT&E. Two excerpts support this approach--"OT&E is essentially an operational assessment of a system's performance where the complete system is tested and evaluated against operational criteria" (AFR 80-14) and "software developed for either new or existing systems shall undergo sufficient operational testing as part of the total system to provide a valid estimate of system effectiveness and suitability in the operational environment" (DODD 5000.3).

The difficulty in planning for OT&E of software lies in determining the extent to which special emphasis should be given to software, and in what areas can (should) software be evaluated separately from the remainder of the system. No set answer applies in all cases. AFTEC groups software OT&E concerns into three areas: performance (how well the software performs its intended function in the operational environment); operator-computer interface (the extent to which the software possesses desirable characteristics from the user's or operator's viewpoint); and maintainability or supportability (how well the software can be maintained in accordance

with the software support concept). Within these areas the approaches to the evaluation vary from program to program. In general, software maintainability can be evaluated without regard to the remainder of the system, whereas software performance and the software operator/ machine interface cannot. Additional discussion on each of these three areas is provided in section II of this handbook.

There is also a need to understand the differences and similarities between OT&E and DT&E. A detailed discussion of OT&E and DT&E is found in AFM 55-43. For the purposes of this handbook, it is sufficient to say that the primary purpose of DT&E is to ensure system compliance with development specifications, whereas the primary purpose of OT&E is to evaluate system capabilities in light of operational (including support) requirements and concepts. However, data from development testing can be used by the operational test agency and vice versa. Operational testers are strongly encouraged to consider the development test data separate from the system developer's evaluation whenever that data can be used to evaluate operational test objectives. Frequently DT&E and OT&E are combined to avoid duplication, shorten test schedules, and reduce resource requirements. The evaluations, however, are always conducted independently.

Software test events may also be combined. The results of specification compliance testing of software may provide valuable information regarding system level performance in the operational environment. Therefore, the software test manager should make sure he knows what software development testing is being performed, understands the nature of the testing, and, when appropriate, make arrangements to attend and/or get results from the testing. Note also that some developmental testing will occur which is not called software development testing, but that will give insight into software operational characteristics. Examples are integration testing (software to hardware and subsystem to subsystem), system testing at various levels (in-plant and in-field), and iterative software development during field test.

C. AFTEC ORGANIZATION FOR OT&E.

The Air Force Test and Evaluation Center was established on 1 January 1974 to fulfill Department of Defense and congressional desires that each of the military services have an operational test organization separate and distinct from their developing and using commands. AFTEC is a test management agency that provides the organizational framework for independently assessing and reporting operational capabilities of Air Force weapon systems.

1. General.

AFTEC is an Air Force separate operating agency reporting directly to the Chief of Staff of the Air Force. The center is comprised of a headquarters located at Kirtland AFB, NM; HQ detachments at remote locations; and field test teams at designated test sites or oprating locations.

The primary mission of AFTEC is to plan and manage the Air Force's operational test and evaluation program. AFTEC plans, directs, controls, and independently evaluates and reports Air Force operational test and evaluation of major programs. These are commonly known as "managed programs." In addition, AFTEC monitors MAJCOM management and conduct of non-major OT&E programs. For these programs, AFTEC approves the test plan and the final report. These programs are commonly referred to as "monitored programs." Thus, the center serves as the principal field command for providing operational test and evaluation information to the Secretary of the Air Force and the Chief of Staff of the Air Force for use in making decisions in weapon systems acquisition programs.

The headquarters staff prepares pretest documentation (such as a test concept), develops test plans, arranges for test resources, assists in data analysis and evaluation, and staffs and publishes the final reports on managed programs. On monitored programs, the headquarters staff (both at Kirtland and at the detachments) provides assistance, as necessary, to the MAJCOMs in the preparation of test plans and final reports; comments on, coordinates, and

approves test plans; and comments on and approves final reports. All AFTEC software test managers are assigned to the headquarters staff at Kirtland AFB, NM.

The headquarters detachments are primarily concerned with specific monitored OT&E programs being conducted at their locations. There are three such detachments: Det 1 at Kapaun AS, Germany; Det 2 at Eglin AFB, FL, and Det 3 at Nellis AFB, NV. Detachment 4 at Kirtland AFB, NM has other responsibilities. At present no software specialists are assigned to the detachments.

AFTEC field test teams are responsible for preparing detailed test procedures, carrying out the test in accordance with the approved test plan, performing the evaluation and preparing the final report. A deputy (or assistant) for software evaluation is normally assigned to the field test team.

Software test managers have frequent contact with personnel from four of the functionally aligned directorates in HQ AFTEC (see figure 1). These are:

Directorate of Test and Evaluation (TE): Through its six test divisions (shown in figure 1), TE exercises overall OT&E management. All test managers and test monitors are assigned to TE. The Software Branch is located within one of the TE divisions, namely the Computer/Support Systems Division (TEB). TE provides the test manager and, when appropriate, a software test manager to the headquarters test team element. Support is also provided by TE and TEB during the advanced planning phase as part of the program planning group.

Directorate of Analysis (OA): Responsible for operational effectiveness test design, data management, and operational effectiveness analysis efforts. Provides an analyst to the headquarters test team element.

Directorate of Plans and Resources (XR): Responsible for advanced planning, OT&E policies and procedures, budgeting, and test resources. Provides a resources representative to the headquarters test team element. Prior to program transition to TE, XR does program planning and chairs a program planning group for advanced planning purposes.

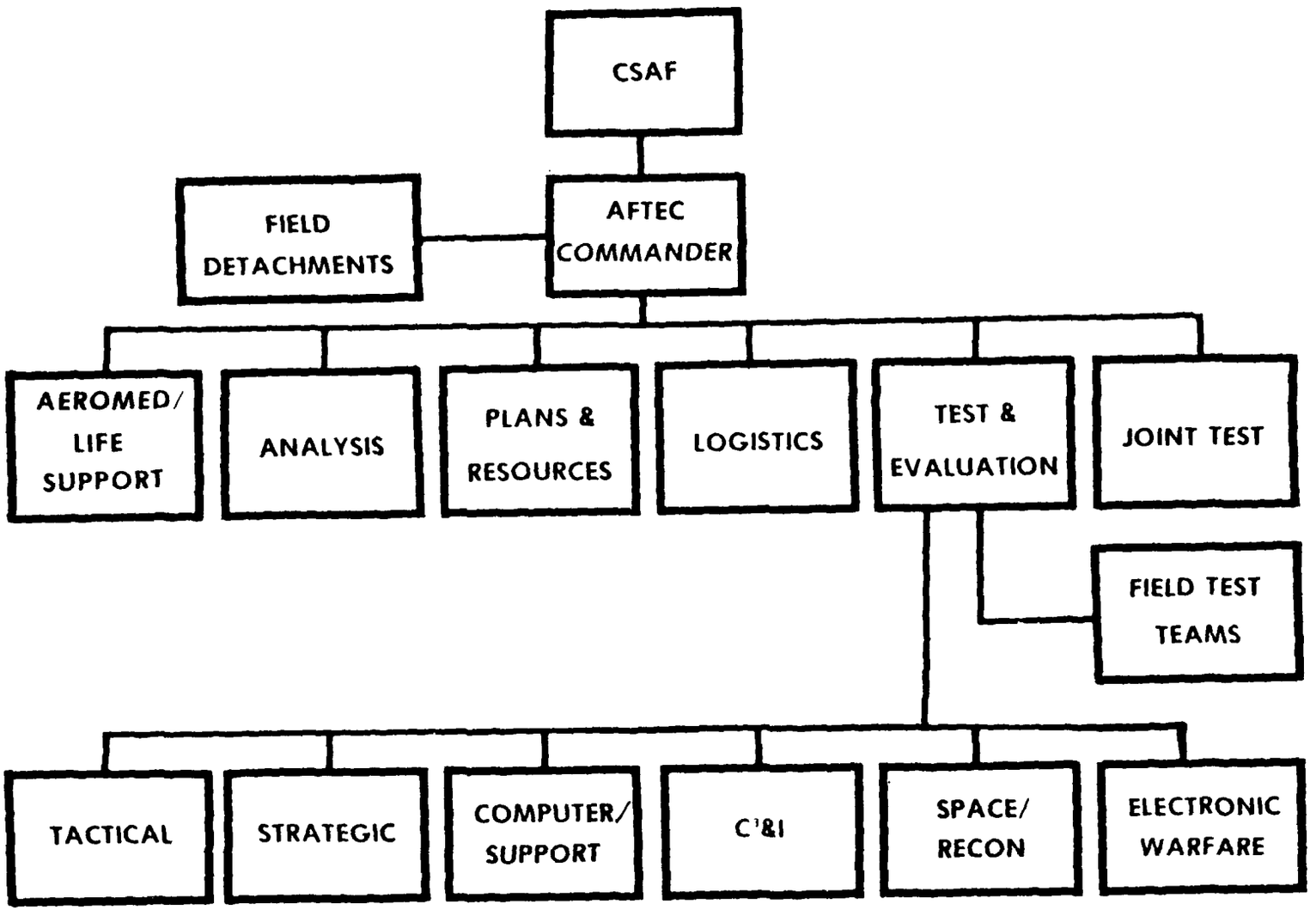

Figure 1. AFTEC Organizational Structure

Directorate of Logistics (LG): Responsible for test design, test planning, and analysis related to reliability, maintainability, and logistics supportability. Provides a logistics representative and a supportability representative to the headquarters test team element.

For a more detailed description of HQ AFTEC organizational responsibilities see AFTECR 23-1.

2. Headquarters Test Team Element.

A group of headquarters personnel (defined in the work directive for each program) is responsible for all phases of test design, test planning, and overall test management. In the early test preparation phases, this group is called the program planning group and is headed by a program manager from XRB. Later, when the program transfers to TE, the group is called the Headquarters Test Team Element. The essential functions are the same.

For test management, the AFTEC test manager draws expertise and support from three directorates - Analysis, Logistics, and Plans and Resources--as well as a safety representative from the Directorate of Safety and a software test manager from the Software Branch (TEBC). These representatives, headed by the test manager, form the Headquarters Test Team Element. Typically, all members of a given test team element, with the possible exception of the test manager, will not be full time on that effort but rather will be on two, three, or more other test team elements as well. On the average, software test managers support three to four different test programs.

The entire test design/test planning function is an evolutionary, iterative process. The process involves definition, evaluation, and refinement of test objectives, measures of effectiveness, and test methodology along with the associated test resources. These items are documented as the planning proceeds. The documents are the test approach, test concept, and the test plan. The required test resources are documented in the Test Program Outline (TPO). The documents are prepared from inputs from the technical specialists (OA, LG, TEBC, etc.), and the information is consolidated by the

program manager (XR) or test manager (TE). Specific guidelines for the software test manager to use in preparing the software test and evaluation information and sections of these documents are provided later in this handbook.

3. Field Test Team.

The field test team is responsible for the actual test conduct in accordance with the approved test plan. This involves preparation of detailed test procedures, scheduling of day-to-day activities, on-the-scene management of test events, and preparation of the final test report.

Typical organization of the field test team is as shown in figure 2. For most test programs, there is a deputy (or assistant) for software evaluation as shown. Normally this individual will be an AFTEC resource as are the test director, deputy for operations, deputy for logistics, and the assistant for data management and analysis. However, on a case-by-case basis the deputy for software evaluation may be provided by the software support agency or the using command. In some rare instances where the test team performs all its duties on a temporary duty (TDY) basis, such as a short duration OT&E, the AFTEC software test manager has served as the deputy for software evaluation; such an arrangement is not recommended.

The deputy for software evaluation is a specialist who heads the group of software evaluators and is responsible for ensuring that all software test objectives are completed and test results reported. Since software has both operational effectiveness and operational suitability aspects and since software may be embedded in both primary mission equipment and in support equipment, the deputy for software evaluation must work closely with the deputy for logistics and the deputy for operations during all aspects of the test. The group of software evaluators typically include some members TDY for short-duration tasks such as completing maintainability questionaires; full-time evaluators to help develop test cases and identify software anomalies and deficiencies; and test team members assigned to

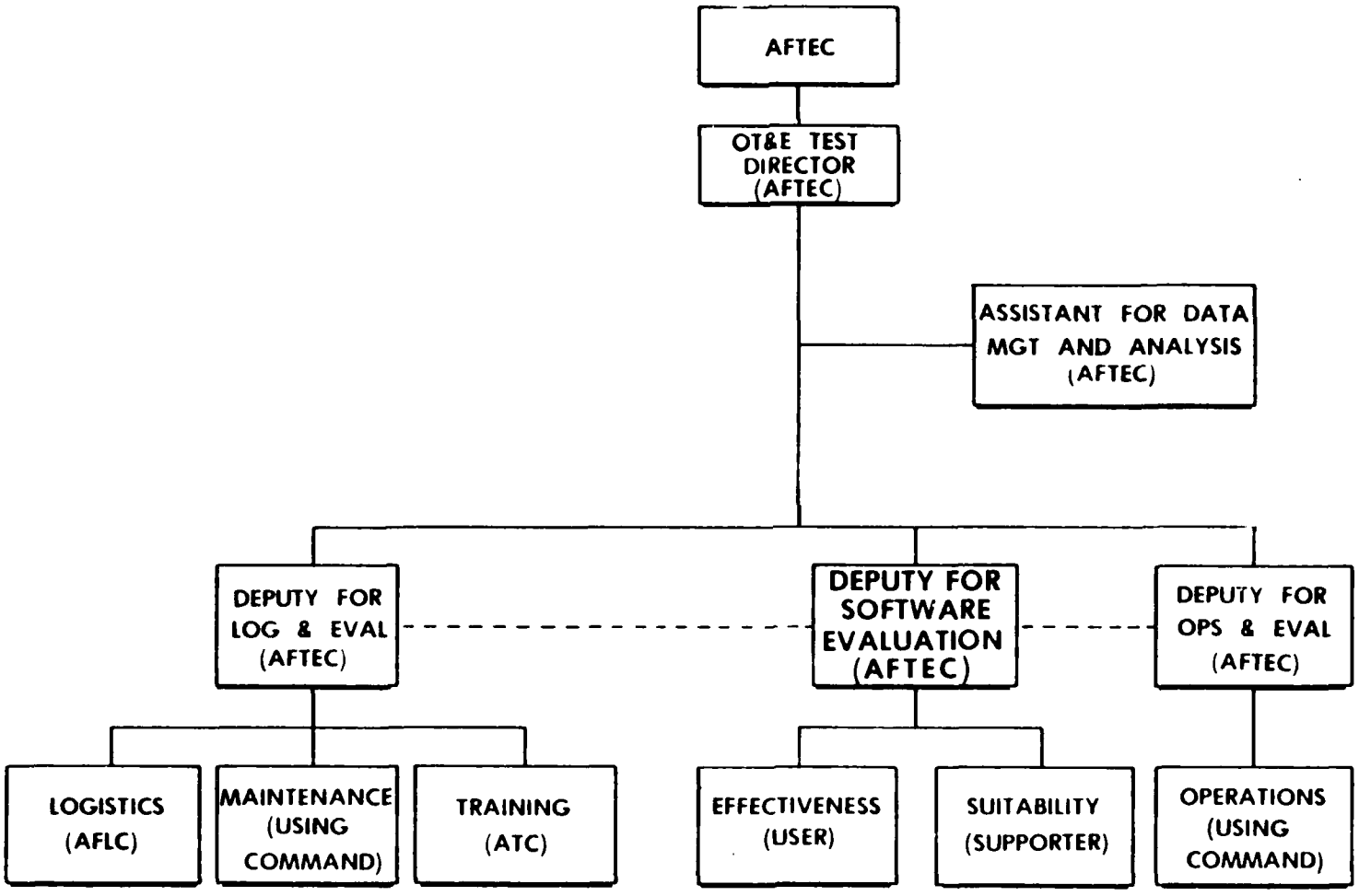

Figure 2. AFTEC Test Management Approach (Test Team)

another deputy, but who provide data for software evaluation. An example of the latter is a group of automatic test equipment operators who work for the deputy for logistics but complete Software Operator Machine Interface Questionaires for the deputy for software evaluation.

Software test managers directly interface with the deputy for software evaluation but there are also many interactions with the other test team members because of the multi-faceted effects that software can have on a modern weapon system.

D. THE OT&E PROCESS: THE SOFTWARE TEST MANAGER'S PERSPECTIVE.

This section outlines the OT&E process and relates the software test manager's functions for each step. Details of each step will be found in section II.

1. Advanced Planning.

This phase is primarily concerned with identification of critical test issues, system test objectives, measures of effectiveness (MOE), and test methodology, and OT&E resource (manpower and cost) requirements. The software test manager identifies objectives and MOEs for the system software and works closely with OA and LG to integrate the software objectives into the overall system objectives. Resource (personnel, equipment, travel, training, etc.) requirements for the test are identified and documented in a test program outline (TPO). The primary documents in this phase are:

a) Test Approach and Test Concept. Prepared by program planning group under XR direction. Coordinated with MAJCOMs as appropriate.

b) Test Program Outlines. Prepared by XR with inputs from the program planning group. Coordination by MAJCOM implies agreement to provide resources as identified.

2. Test Planning.

This phase is an evolution from previous planning. The HQ Test Team Element refines earlier efforts, specifies data requirements, ensures the scope of testing is correct, establishes evaluation criteria, etc.

The primary product of this phase is the IOT&E test plan. The software test manager participates, with other directorates, as a member of the HQ Test Team Element to ensure software concerns are adequately addressed.

3. Test Conduct.

The HQ Test Team Element is responsible for continued monitoring of the field test team progress and ensures that the interpretation of the test plan is correct. Some data reduction and preliminary analysis at headquarters may be accomplished. The software test manager works with the deputy for software to provide necessary guidance or assistance.

4. Final Report.

This report is the responsibility of the test director. The HQ Test Team Element assists the test director and coordinates the report through HQ AFTEC.

5. Other Activities for the Software Test Manager.

In order to ensure that software issues are adequately addressed, the software test manager will also participate in various working groups and attend design reviews. Two working groups merit special mention:

a) Test Planning Working Groups (TPWG). This group is chaired by the System Program Office and consists of members from all organizations involved in the testing--development and operational. This group

prepares the Test and Evaluation Master Plan which is the integrated plan for all testing and defines the required test resources.

b) Computer Resources Working Group (CRWG). This group is comprised of members of the using command, supporting command, and acquisition agency. Chaired by the System Program Office, the principle product is the Computer Resources Integrated Support Plan (CRISP). The CRISP provides a life-cycle plan for management of the software.

SECTION II

SOFTWARE OT&E CONCERNS AND TECHNIQUES

A. INTRODUCTION.

The software test manager generally focuses his attentions on the following areas: software performance, software operator-machine interface, support system effectiveness, and software maintainability. This section will outline each of these areas and discuss tools/techniques available that the software test manager can specify for the test.

B. SOFTWARE PERFORMANCE.

The effectiveness of software in its operational configuration is difficult to quantify. There are no metrics developed that characterize the nature of software performance or availability. Thus, in this area, the software evaluation focuses on software problems that arise during system operation and the effect the problems have on the system. The software test manager and the deputy for software evaluation should attempt to define test scenarios for the system that maximally stress known or suspected weak spots in system design. When an independent verification and validation contractor exists, he should be called upon to analyze test results and/or identify possible test scenarios. An event trace monitor (ETM) can be used to assess timing margins and to provide data for assessing other objectives (e.g., response times under various conditions).

Software performance evaluation (from an OT&E standpoint) is always done within the context of overall system performance in an operational environment. Note that DOD Directive 5000.3 states that "performance objectives and evaluation criteria shall be established for both full-system and casualty mode operations. For embedded software, performance objectives and evaluation criteria shall be included in the performance objectives and evaluation criteria of the overall system."

Currently, the OT&E test teams are using the reporting methods detailed in TO 00-35D-54 to report all observed system deficiencies. The OT&E test team's deputy for software evaluation reviews anomolies and problems prior to the declaration of a deficiency to determine the contribution of the software to the observed deficiency.

When the system analysis of a problem report indicates that software is associated with the problem, a software evaluator is assigned to further investigate the deficiency. This investigation will include preparing a computer program observation report (CPOR) (appendix 3) on the problem and recommending retest and cerification of any corrective actions. The report is tracked within the test team, and a service report (SR) is written in accordance with TO 00-35D-54 if the CPOR does indeed indicate a problem worth tracking. The SR system then tracks the problem until corrected, at which time an administrative action can note the correction on each related CPOR.

C. SOFTWARE/OPERATOR INTERFACE.

The nature of the operator interaction with the computer is assessed to ensure that adequate consideration was given to the design of this interface. Typical areas of interest are range of response, degree of protection, understandability, flexibility, etc. This area is assessed through the use of standard questionnaires. Operators are asked to complete the questionnaires while operations are fresh in their minds. The results are quantitatively evaluated and performance characteristics of the interface assessed. Volume IV of these guidelines gives details and includes the questions.

D. SUPPORT SYSTEM EFFECTIVENESS.

This evaluation addresses the capability of the software support system to support the software maintenance team. Efforts are currently underway to develop a methodology and test tools for assessing the adequacy of the support system. In addition,

pertinent documentation is reviewed for adequacy, and, when practical, hands-on use of the facility is tried and operator's subjective assessment made of its effectiveness.

E. SOFTWARE MAINTAINABILITY.

This evaluation focuses on the quality of the computer program code and supporting documentation. A representative sample of modules is selected and thoroughly evaluated using a standard questionnaire. The details of the evaluation are contained in volume III of these guidelines.

F. STANDARD QUESTIONNAIRES.

Standard questionnaires currently exist for software maintainability and for the software operator-machine interface. These questionnaires are in volumes III and IV of those guidelines, respectively.

1. Operator-Machine Interface Evaluation.

a. General.

In the past, the operator-machine interface for computer-based equipment has been evaluated on an exception only basis; i.e., each user or operator would comment only on those areas of the interface that particularly disturbed him. Operators would simply rate the interface "good" or "bad" according to the number and difficulty of the problems they encountered. This method of analysis naturally resulted in highly subjective, nonspecific results. Futhermore, one would expect experienced operators to have less problems than inexperienced operators merely because they know the system peculiarities.

Highly subjective evaluations are undesirable because they often yield questionnable estimates of operational capabilities and do not sufficiently describe specific problems that need to be fixed to increase operational capabilities.

b. Evaluation Techniques.

In an effort to decrease the subjectivity of analyses of (the software portion of) operator-machine interfaces, AFTEC has developed the software operator-machine interface questionnaire (SOMIQ). Each operator/evaluator is guided via questions to isolate and consider a number of quality factors about the equipment being evaluated. Through this organized approach, the operators all consider the same aspects of the operator-machine interface, thereby yielding a consistent analysis. Furthermore, the operators are guided to consider subjects which they might overlook if asked to prepare lists of problem areas. Additionally, information is obtained about which aspects of a system contribute positively to operational capabilities. The questionnaire consists of questions addressing various aspects (factors) of assurability, controllability, workload reasonability, descriptiveness, consistency, and simplicity.

Details on using the SOMIQ, and the questionnaire itself, are found in volume IV of these guidelines, The Software Operator Machine Interface Evaluator's Handbook. Keep in mind that this is not a human factors evaluation per se - we are not interested in questions like how much glare is on the screen, or how comfortable the chair is. We are concerned here only with the operator communicating with the system via the software.

c. Methodology.

Many systems that lend themselves to evaluation by a tool such as the SOMIQ have several different stations or applications that could be evaluated. For example, an aircrew training device may have one station for a pilot, a second for an electronics warfare officer, a third for the simulator maintenance technician, and a fourth for the instructor operator. The software test manager must decide which stations he wishes to evaluate, and then arrange for qualified operator personnel to complete the questionnaire for that station.

Since there is but one SOMIQ completed per evaluator per station, it is often cost effective to not digitize the data, but enter it manually. In any event, once the data has been entered into a file, TEBC has a computer program that can reduce the data and perform some statistical analysis. The results of the analysis can then be compared to evaluation criteria of the test plan.

Not only does the software test manager need to select the operator positions to be evaluated, he must also determine what functions are included in a single question, when not to use the questionnaire, etc. The software test manager needs to prebrief the questionnaire carefully with the evaluators, realizing that these are operators who may have little software knowledge, and thus any questions on terminology need to be cleared up before the evaluation proceeds.

2. Software Maintainability.

a. General.

Software maintenance is an activity performed to change a computer program, whether it be to remove errors, to add or delete features, or to modify the program to be compatible with a hardware change. The minimum resources required to design and accomplish a software change include the software source listings, narrative documentation for the software, and computer support resources required to accomplish and test the change (figure 3). AFTEC has designated these minimum resources as separate categories to be examined during software maintainability evaluations.

b. Evaluation Techniques.

(1) Documentation and Source Listings.

In the past the approach to evaluating software documentation and source listings has not been quantified. Typically, one or more knowledgeable persons would examine the documentation and source

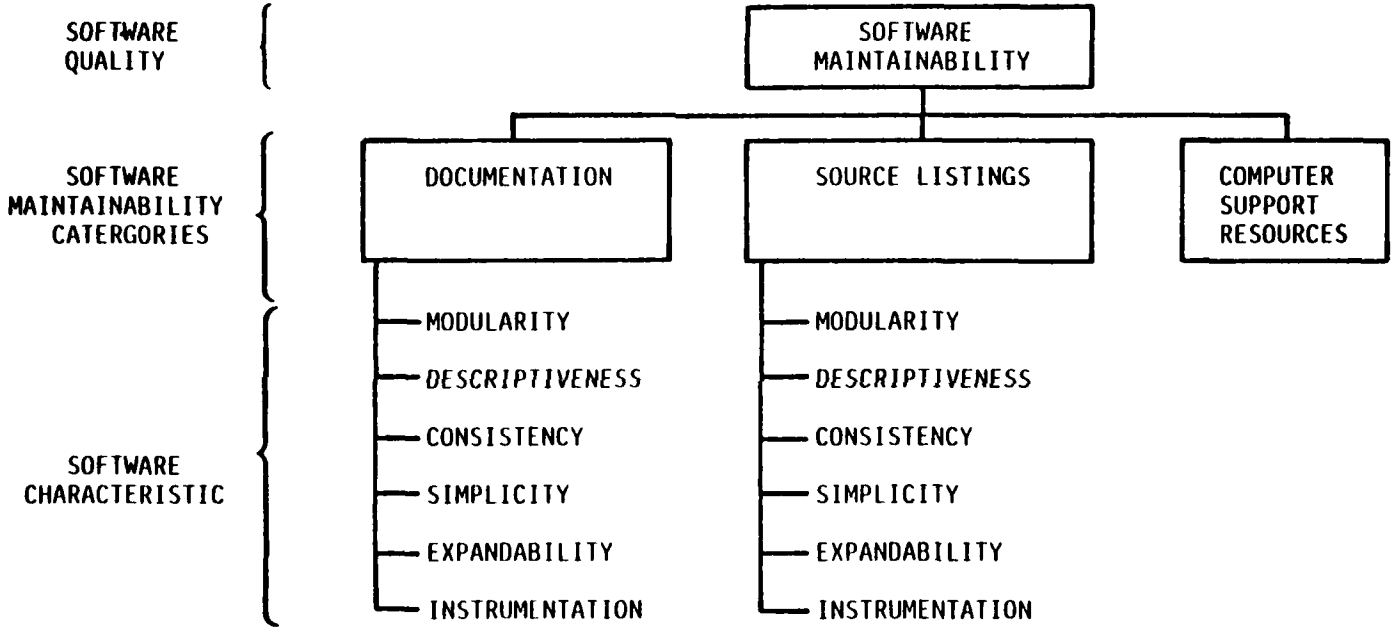

Figure 3. Elements of Software Maintainability

listings and provide a subjective appraisal to an interviewer who, in turn, would make his own subjective interpretation of the evaluator's remarks. Unfortunately, the evaluator's thought processes were not guided to specific considerations and the same criteria may not have been used by each of the several evaluators. Additionally, the interviewer and the evaluator may or may not attach the same meaning to terminology and may or may not assign the same importance to a given factor. The net result of such an evaluation tends to be what may be summarized as "one-word" test reports--good, bad, acceptable, marginal. Regardless of the accuracy of these adjectives, they do little toward highlighting specific strengths or weaknesses. For example, a particular software program may be acceptable overall but may be lacking in documentation, or the documentation may be sufficiently descriptive but not easily understood. Without a consistent, organized method of evaluation which can be applied across a broad spectrum of software programs and programmers/evaluators, establishing creditable evaluation criteria is extremely difficult.

Several studies and tests were conducted so that AFTEC personnel could address the problems of establishing a consistent, organized method of software evaluation which yields creditable evaluation results. The approach that AFTEC has developed, tested, and applied in software OT&E is the use of closed-form questionnaires. The approach requires several evaluators to rate various maintainability considerations of software on a multipoint scale. Appendix 6 of this handbook is description of this approach.

(2) Computer Support Resources.

The computer support resources are those required to perform software maintenance. These resources include the computers and associated supporting software, physical plant, personnel, training, maintenance procedures, test tools, distribution resources, and hardware and software documentation required to accomplish, test, and implement a software change. Frequently, the resources used during software development and integration, or very similar resources, are proposed for software maintenance.

Support software and procedures are the basic tools for software maintenance. They provide the capability to produce executable computer programs with the desired modifications and the capability to accomplish limited testing of small portions of the computer program. Support software includes compilers, assemblers, automated configuration management aids, and interpretative computer simulation software. Support procedures include formal and informal published techniques for updating software and maintaining configuration management.

c. Methodology.

(1) Documentation and Source Listings.

The AFTEC software documentation and source listings evaluation methodology consists of having five or more software evaluators complete standardized, closed-form questionnaires for each computer program evaluated. The two questionnaires used are the software documentation questionnaire and the module source listing questionnaire. The evaluators themselves should be personnel equivalent in backgound and qualifications to those who will eventually maintain the software.

Each evaluator completes the software documentation questionnaire for each computer program evaluated. The questionnaire consists of questions which when completed, provide a measure of the extent to which the software design, as reflected in the documentation, possesses good maintainability characteristics. In addition, information is gathered on the format and organization of the software documentation.

The modules considered in the evaluation are assumed to be representative of the complete set of computer program modules. A random sample of the software subroutines or modules is selected by the deputy for software evaluation, in conjunction with the evaluators, and as approved by the software test manager. Each evaluator then completes a module source listing questionnaire for each of the selected modules. The questionnaire consists of

questions which, when completed, provide a measure of the extent to which the module source listings reflect a software implementation with good maintainability considerations. In addition, the module source listing questionnaire contains questions that will be used to evaluate the consistency between software documentation and the source listings.

This test methodology requires a minimum of five evaluators knowledgeable in software procedures, techniques, maintenance, and the general programming language of the software to be evaluated. Five evaluators are necessary for statistical confidence that the test data provides a valid measure of software maintainability. As a first step in the evaluation, the evaluators are briefed on the questionnaires. Then a trial run is conducted wherein each evaluator completes one software documentation questionnaire and a module source listing questionnaire. Following the trial run, a debriefing is conducted by the AFTEC Software Test Manager to resolve any uncertainties among the evaluators in their understanding of the questions. The questionnaires for the remainder of the selected modules are then completed, usually two to three modules being evaluated per day.

Although the questionnaires require a response to each question, evaluators are encouraged to provide written comments/ expanded narratives as appropriate or desired.

The "questions" themselves are actually positive statements relating to desirable maintainability characteristics of software. Examples are "This module contains checks to detect possible undefined operations," and "Variable names are descriptive of their funtional use." The evaluator is required to mark one of six responses, ranging from "completely agree" down to "completely disagree." To obtain a quantitative result from the responses, each response is assigned a numerical value of one to six points, with six being the highest (completely agree) and one the lowest (completely disagree).

The questions are grouped according to factors (such as modularity, descriptiveness, etc.) whose presence or absence in source listings or documentation directly affects the software's

maintainability. For each factor, an average score is calculated from the responses and is then multipled by a predetermined relative weight (importance). The weighted scores of the factors are then summed to obtain an overall score for the maintainability category being examined, i.e., documentation or source listing.

The resulting score for software source listings or documentation may again be weighted to obtain a higher component of the maintainability assessment. At any level, a score may be compared to predetermined evaluation criteria (goal, standard, and threshold) to identify a possible problem area for futher investigation or to identify an unsatisfactory condition which requires improvement.

Details on using these questionnaires, and the questions as well, are found in volume III of these guidelines, The Software Maintainability Evaluator's Handbook.

(2) Computer Support Resources.

Currently, resources that are sufficient to perform software maintenance are very difficult to evaluate because they are seldom available during OT&E. In addition, AFTEC has not yet developed a comprehensive, measurable list of factors to be included in the evaluation. Therefore, the computer support resources evaluation is quite subjective and often consists of a review of software support plans and can be highly dependent on inputs from the supporting agency. AFTEC is hopeful of achieving improvements in evaluating computer support resources. One promising solution involves use of support agency personnel to perform IV&V (see paragraph H below) and to develop and use the eventual support facility to conduct the IV&V. The benefits to be realized from implementing this approach are that (in addition to standard IV&V benefits) the software support facility is available for evaluation during OT&E and earlier organic support of softwareis possible.

Details on the AFTEC approach to evaluating computer support facilities will be published in volume V of these guidelines, Computer Support Resources Evaluator's Handbook.

G. EVENT TRACE MONITOR (ETM).

1. General.

AFTEC has been applying an event trace concept to OT&E on a limited basis since 1978. The principal capability of the event trace monitor (ETM) is monitoring the processing performed by a computer and recording the occurrence and time of key events. With this capability and post-test data reduction, the ETM can be used during OT&E to determine the amount of reserve processing time available for future additions and to verify that the processor is not on the verge of malfunctioning because of stressed operating conditions. It is also possible to investigate failures which occur during OT&E and to determine if failures were caused by hardware or software. In order for the ETM to provide useful data for OT&E, it must monitor, in real-time, the operational program during execution in the operational processor(s). Ideally, this monitoring would occur in a realistic operating environment (e.g., inflight for an avionics system). This may require a pod-mounted ETM or at least a flight qualified configuration. AFTEC does not currently have this capability. Alternatively, the ETM could be used in a simulated mission environment which would include, for example, the flight processor loaded with the OFP and a realistic simulation of the external environment. AFTEC has conducted testing similar to this alternative with the F-16 fire control computer and the F-16 independent assessment simulator at the Air Force Avionics Laboratory, Wright-Patterson AFB, Ohio. Results of this testing and additional technical details on the ETM are available in the AFTEC/TEBC publication, Applications of the Event Trace Monitor to Software Operational Test and Evaluation (May 1980)).

2. Methodology.

Use of the ETM requires considerable initial investigation of the software to determine which addresses must be monitored to provide the necessary data for problem solving. The addresses ideally would

each relate to a particular significant event in the execution of the program, i.e., when the address match occurs, the event is known to have taken place. A second type of event could be represented as a "signal edge," i.e., an event has occurred when a signal increase or decrease is monitored by probes. Thus, before any monitoring session the user must "program" the control memory of the ETM. This procedure is accomplished by entering the predetermined addresses (in octal format) into the data handler control memory of the ETM, using its front panel switches. The control words select the comparator limits, the data paths, the operation performed by the data functions, the time-stamp clock interval, and other control functions.

The output of the ETM is a nine-track magnetic tape recording of the history of the requested events, each marked with a time-stamp to indicate when they occurred. This data can then be reduced by the software test manager, using the AFTEC developed data reduction package. A user's manual provides information on how to install the package, how to use the program, and how to interpret program outputs. This manual is available through AFTEC/TEBC.

H. SOFTWARE INDEPENDENT VERIFICATION AND VALIDATION.

1. General.

As one way of addressing the many years of cost overruns, slipped schedules, and unreliable end products associated with the development of software embedded within an operational weapon or information system, procuring agencies of the Department of Defense have instituted a policy of contracting with a company that is separate and distinct from the developing company to oversee the design, development, and test of a system's embedded software. This overseer role is referred to as Software Independent Verification and Validation (IV&V). Although the requirement for IV&V is becoming a standard part of all major system full scale development contracts, the scope of the IV&V activities vary substantially from one system to another.

At one end of the spectrum is a monitoring activity consisting of specification reviews, performance analysis using simple models of the software and computer system, and test data analysis to ensure specification compliance. This is an inexpensive approach to IV&V, but it may not contribute substantially to any significant increase in the quality or reliability of the software subsystem; nor does it affect to any great extent the cost or schedule of the software development effort.

At the other end of the spectrum is an IV&V activity that will cost as much as 50 percent of the primary software development effort. Under this approach, the IV&V contractor will:

a) Independently derive all major algorithms incorporated in the software and perform detailed performance analyses on these algorithms.

b) Construct a detailed model of the software and computer subsystems and conduct detailed studies of these subsystems under various loading conditions and operational scenarios.

c) Perform detailed reviews of all software specifications and other documentation and conduct requirements traceability analyses to ensure completeness and accuracy of all requirements.

d) Review all code for accuracy, efficiency, and conformance to coding standards.

e) Generate an independent set of test procedures for software validation and conduct tests in accordance with these procedures using an independent test facility.

f) Repeat tests conducted by the development contractor using an independent test facility and compare results with development contractor.

Although this approach may well result in a highly reliable and high quality software subsystem, it is a very expensive approach in terms of more than dollars. Because of the extensive in-line analyses performed by the IV&V contractor, some of the development contractor's work must wait on completion of tasks by the IV&V

contractor. As a result, this approach often results in cost and schedule overruns in excess of what might have resulted from the development contractor's efforts alone.

The majority of the IV&V efforts conducted on systems in full scale engineering development lie somewhere between these two extremes and perform a very useful and vital role in ensuring the quality and reliability of the end product. Because of the relative newness of the IV&V concept, it will take some time and some experimentation to determine the point in the spectrum that is most effective in quality/reliability assurance and cost/schedule performance.

In order to minimize difficulties, ensure independence, and prevent anomosity or an adversary relationship, the program office will have to enforce management controls to maintain an effective IV&V effort. Such controls require the contractors to communicate through the program office rather than directly on specific aspects of the software. The program office may authorize irect contact between certain individuals or on certain subjects, such as policies or use of test tools. The software test manager must assist the program office in fostering a spirit of cooperation and partnership in this delicate relationship. Handled correctly, IV&V pays dividends beyond its original cost.

2. OT&E Benefits from IV&V.

The data and results from IV&V can be used to identify stress points for additional testing, to identify candidate scenarios, to key the evaluators to trouble spots, etc. It is recommended that IV&V contracts include provisions for communication with the OT&E test team, for identifying candidate system level tests, and for participating in the analysis of the OT&E test results.

I. DUTIES OF THE SOFTWARE TEST MANAGER.

1. Advanced Planning.

The software test manager should get involved as early as possible on any software intensive program. This gives him more time to understand the software which will facilitate evaluation, and more importantly, to accomplish the planning for adequate test and evaluation of the software. It is for this purpose that the software branch established a position for advanced planning, thus providing a focal point for close coordination with AFTEC/XR. The TEBC advanced planning representative will provide the link between the XR advanced planning function and the eventual software test manager for the program. The TEBC advanced planner's primary duties are participating in the advanced planning process for new programs to ensure the software evaluation is properly integrated with the overall system as reflected in the test approach and test concept.

As implied by the above, the software test manager's duties begin much earlier than formal transfer of the program from XR to TE. Many of these early duties are accomplished by the TEBC advanced planning manager. Following are the duties of the software test manager.

2. Initial Duties.

The software test manager establishes the initial data base of knowledge for the software on a program. He will:

a) Review available documentation to become the "resident software expert" on the system and its software.

b) Review the contract data requirements list (CDRL) early to ensure the appropriate software data items will be available and delivered to AFTEC or the test team.

c) Serve as a repository of early software development information that can be passed on to the test team. (This might mean establishing an initial software management information system.)

d) Interface effectively with the program manager and development contractor for those situations requiring "persuasion" to obtain informal or undelivered data. (This responsibility for establishing good working relationship is critical).

e) Work with the TEBC advanced planner to ensure early involvement of software data "thinking" (in the OT&E approach if necessary).

f) Attend key progress reviews (PDRs, CDRs) to understand design and operational issues (often used as the basis for test design).

g) Establish training requirements and courses for the software analysts and evaluators on the test team. This may include writing the training course description, course outline, and learning objectives for the software evaluators.

3. Liaison Duties.

The software test manager may well be the only software OT&E advocate working with the program office and contractor(s). His planning and diplomacy may be the only tools he has to obtain software data, activities, and even a mutual understanding of what AFTEC software OT&E involves.

This activity is critical. Without contractor and program manager cooperation, the OT&E software evaluation can evaporate. This is particularly true if the program manager is not cooperative and serves as an obstacle between AFTEC and the contractors.

Although easier to accomplish, it is also important that the software manager effectively interface with all other agencies, e.g., COMOPTEVFOR, OPEVAL, IV&V contractors, using and supporting agencies, etc.

4. Formal Test Planning.

The software test manager and the software advanced planner establish the guidelines and charter for all future evaluations of that program's software.

This formal test planning involves more than just writing the OT&E approach and test plan. It includes attending meetings (detailed later under paragraph L) and initializing the liaison and data repository functions identified above; it includes setting the framework for the software evaluators on the test team.

As stated previously, system testing can mask the appropriate software test needs. An important function of the software test manager and the software evaluators on the test team is to identify the critical software functions and to ensure that the test scenarios adequately exercise the critical functions.

5. Choosing the Deputy for Software Evaluation.

The final activity is ensuring that the deputy for software evaluation (DSE) on a program is knowledgeable, creative, and eager. Often the DSE is identified and assigned because he was the first 51XX or 28XX available for reassignment. Since this is an unacceptable solution, the software test manager must ensure that the DSE has the necessary qualifications.

Ideally, once the test team is up to speed, the software test manager will serve primarily as an advisor and participant in higher level management issues. Further, he will want to ease the DSE into the liaison role. Thus, the DSE should be capable of managing the software evaluators, conducting future planning activities for the program, suppervising all software evaluations, and preparing the software portions of the test report.

If the DSE does not have these capabilities, the software test manager will find himself in the position of trying to be both software test manager and DSE.

J. GOVERNING DOCUMENTS.

There are a number of Department of Defense directives and Air Force and AFTEC regulations and manuals with which the software test manager will become involved. Following is a description of those that would head any such list.

1. Department of Defense Directive 5000.3.

(DODD 5000.3 - Test and Evaluation.) This directive is the real basis for test and evaluation in the Armed Forces. It establishes all policy for the conduct of test and evaluation in the acquisition of defense systems (including everything from major ships of a class to computer software). There is, in fact, a separate paragraph 6 that addresses Test and Evaluation of Computer Software. Four important excerpts from this paragraph are as follows:

a) "Performance objectives ...shall be established for software during each system acquisition phase."

b) "Decisions to proceed from one phase of software development to the next will be based on ...appropriate T&E."

c) "...software shall undergo operational testing ...utilizing typical operator personnel."

d) "The OT&E agencies shall participate in software planning and development to ensure consideration (of the) operational ...environment and early development of operational test objectives."

In addition to this general guidance, the enclosures to 5000.3 contain definitions and guidelines for the test and evaluation master plan.

2. Air Force Regulation 80-14.

(AFR 80-14 - Test and Evaluation.) This regulation is the Air Force's implementation of DOD Directive 5000.3. It gives the Air Force policy and procedure for managing test and evaluation

activities of defense systems in the Air Force. It also establishes the management relationships during OT&E between AFTEC, the implementing command, and the operating and supporting commands. The regulation specifically applies to software systems, subsystems, and components. Paragraph 8 addresses computer software test and evaluation, and expands upon those software T&E principles contained in DOD Directive 5000.3. In addition, independent verification and validation (IV&V) is advocated, and definitions for IV&V terminology are provided in the glossary.

3. Air Force Regulation 800-14.

(AFR 800-14 - Management of Computer Resources in Systems.) AFR 800-14 is in two volumes. Volume I establishes the policy for the acquisition and support of computer equipment and computer programs, and volume II details the procedures for implementing that policy. Volume I is very short, but does list some definitions of interest to the software test manager. Volume II contains details on the acquisition process. An example is paragraph 2-8, Computer Program Development in the System Acquisition Life Cycle, which gives the basic "waterfall" acquisition cycle chart for software of analysis phase, design phase, coding and checkout phase, test and integration phase, installation phase, and operational and support phase. Volume II also describes directives and plans, including the Program Management Directive (PMD), the computer resources integrated support plan (CRISP), and the computer resource working group (CRWG). Verification and validation is discussed, as well as configuration management of computer resources (including operational/support configuration management procedures).

4. Air Force Manual 55-43.

(AFM 55-43 - Management of Operational Test and Evaluation.) This manual also is in two volumes. Volume I provides the general guidelines on planning, managing, conducting, and reporting on OT&Es, whereas volume II contains the specific procedures or

techniques needed to address day-to-day problems and questions on the conduct of OT&E. Again, volume II will be of most help to the software test manager. This volume provides standardized formats for test program outlines, test plans, test reports, etc. It also provides discussions of those items such as operational effectiveness and operational suitability that must be considered when specific test objectives are written, and checklists that highlight essential elements to be accomplished during the various phases of OT&E. The chapters in volume II are essentially examples that correspond to the general information in the corresponding chapter of volume I. Although a specific software annex is not exemplified, the general format of the test plan is provided in annex 8-9 of volume II.

5. Air Force Test and Evaluation Center Regulation 55-1.

(AFTECR 55-1-AFTEC Operations Regulation.) AFTECR 55-1 outlines how AFTEC does its job and could be called "The AFTEC Test Manager's Handbook." It describes test planning, test director responsibilities, OT&E reports, AFSARC and DSARC review boards, the test program case file, and the OT&E Management Information System. Formats and samples are given for test plans, data management plans, etc. AFTECR 55-1 should give the software test manager a good overview of the AFTEC mission in action.

It is in AFTECR 55-1 that the concept of a single OT&E test team structure is expounded. This test team structure consists of a headquarters element and a field element, as was described in these guidelines under AFTEC organization.

K. IMPLEMENTING DOCUMENTATION.

There are a number of program documents with which the software test manager must be familiar to determine if software testing concerns have adequately been considered in the design, development, and implementation of the system.

The following is not intended to be an all-encompassing list or a complete description of such documentation but rather to serve as an introduction and checklist for the software test manager.

1. Statement of Operational Need (SON).

The using command establishes a requirement for a particular capability in a statement of operational need (SON). This would normally be the earliest statement of overall operational requirements against which the system, including software, will eventually be tested. The software test manager should not expect to see a SON on all programs, nor should there be much on software in a SON. Basically, the SON establishes short falls in existing capabilities and the improvements required to resolve those short falls. The format for a SON is in AFR 57-1; a mission element need analysis (MENA) is usually submitted as an attachment to the SON. HQ USAF then uses this information to prepare a Mission Element Need Statement (MENS).

2. Program Management Directive (PMD).

Acquisition and modification programs receive Air Staff direction and guidance in the form of the program management directive (PMD). The PMD is prepared by the Air Staff program action officer, or the program element monitor (PEM). It governs the actions and participation of the implementing, using, supporting, and other participating commands in the program. A detailed format for the PMD is found in AFM 55-43, vol II, annex 6-2. AFTEC reviews and comments on PMDs to ascertain that proper OT&E provisions have been included. A checklist for AFTEC review of PMDs has been compiled by HQ AFTEC/XR and is provided as appendix 1. Overall procedures for AFTEC review of PMDs are provided in a 23 October 1978 AFTEC policy letter. The software test manager should check to see that software independent verification and validation has been addressed; if it has not, he should suggest in his review uses for this procedure.

3. Test and Evaluation Master Plan (TEMP).

The TEMP is a formal document required by OSD and approved by them at each milestone (see DOD 5000.3, paragraph 9). The

TEMP is not a tasking document, but an agreement between the program manager and the test participants on the scope and requirements of the program and the roles of the participants. It is prepared by the members of the test plan working group (TPWG, defined in paragraph L2), approved by the program manager, and coordinated by the headquarters of all planning participants. A detailed format for the TEMP is in AFM 55-43, vol II, annex 6-3. In addition, TEMP guidelines are at enclosure 2 to DOD 5000.3. The software test manager should review the TEMP to see if software concerns have been adequately addressed. If IV&V is to be employed on the program, the TEMP should outline the extent of the IV&V and the tasks that are expected to be performed by the IV&V organization. Appendix 8 to this handbook is a sample TEMP review.

4. Computer Resources Integrated Support Plan (CRISP).

From the viewpoint of the software test manager, the CRISP is one of the more important pre-production documents; unfortunately, it is not always available in a timely manner for OT&E.

The CRISP identifies organizational relationships and responsibilities for the management and technical support of computer resources. It also identifies documentation, an important part of our evaluation. It is prepared by the members of the computer resource working group, (CRWG, defined in paragraph L3), approved by the program office in conjunction with the supporting and using commands, and coordinated by the appropriate commands. A detailed format for the CRISP is provided in AFLCR 800-21, attachment 2, 4 January 1980 (appendix 2). As an associate member of the CRWG, AFTEC does not approve or disapprove CRISPs, but the software test manager should review the CRISP to ensure that all embedded computer systems are described, support concepts are defined, any software maintenance facilities are discussed, and logistics and special provisions such as handling of firmware (ROMs, PROMs, etc.) have been addressed.

5. Operational/Support Configuration Management Procedures (O/S CMP).

The basic configuration management approach contained in the CRISP will be detailed further in the O/S CMP, written by the supporting and using commands in conjuction with the implementing command. The O/S CMP includes the provisions for change control and other procedures as outlined in AFR 800-14, vol II, para 6-10. AFTEC does not approve/disapprove O/S CMPs, but the software test manager should review the O/S CMP to see if it goes beyond the CRISP in describing processes and procedures to accomplish software maintenance. In particular, check to see that the facilities and equipment to be provided match the support procedures. An example might be a CRISP that calls for annual updates of an EPROM; the O/S CMP should be checked to ensure that a field level reprogramming capability has indeed been provided for. In general, the O/S CMP is prepared later in the program than the CRISP, as it is only due before the PMRT. Although it should be developed well ahead of that milestone, the timing may preclude the AFTEC software test manager from obtaining an O/S CMP for review for OT&E purposes.

6. Test Program Outline (TPO).

The test program outline, prepared by HQ AFTEC/XR in accordance with AFM 55-43, is basically a resources document, not a tasking document, that quantifies test support requirements for OT&E. It is sent to each agency that will support the OT&E project, and action addressees respond with their intent of support (or nonsupport) to the requirements. The actual OT&E program requires an approved PMD or test directive (TD). The TPO is updated semiannually, and following each update the Operational Resource Management Assessment System, Test and Evaluation (ORMAS) colonels' group meets to resolve any conflicts. The ORMAS/T&E is chaired by the Directorate of Operations and Readiness, HQ USAF, with participants from the MAJCOMs and agencies involved in the

support and conduct of T&E. Agreements made between ORMAS/TE participants are considered binding provided funding and HQ USAF direction (PMD, TD, etc.) is issued. The TPO is then published as a P-series (PO) document under the "USAF Program for OT&E."

The software test manager must review and coordinate on the TPO to ensure that provisions have been made for a deputy for software evaluation and the required software effectiveness and suitability evaluators.

7. Other Documents.

There are, of course, other implementing documents that can be of benefit to the software test manager. The maintenance concept and the operations concept, as defined in AFR 57-1, are two such document that can be a valuable source when available. Also, the program's systems specifications (A-level) and derived specifications (B-level) will provide invaluable insight.

L. MEETINGS.

An AFTEC Commander once summed up any test manager's job in three directives:

a) Know the people.

b) Know the process (of acquisition).

c) Know the program.

All of these actions can be accomplished in various ways, but one of the most direct is through meetings of the participants in a program. This section lists some of the meetings that should be attended by a software test manager.

1. Advanced Planning Meetings.

Advanced planning is accomplished within HQ AFTEC by XR in conjunction with representatives from OA, LG, and TE (including the Advanced Planning Software Manager from TEBC). It consists of meetings within HQ AFTEC to gather information prior to briefing

the directors on a preliminary OT&E test approach for a program. As detailed in 1-1, TEBC has an advanced planning individual assigned to cover these meetings to keep the branch chief and, eventually, the software test manager informed.

2. Test Planning Working Group (TPWG) Meetings.

The Test Planning Working Group (TPWG) is formed and chaired by the implementing command (usually AFSC). Membership is drawn from the program office, applicable AFSC test agencies, the OT&E command, the using command, supporting commands, and (when appropriate) contractors. The primary output of the TPWG is the Test and Evaluation Master Plan (TEMP), but the TPWG also performs several other useful functions. In general, it provides a forum for test and evaluation subjects, to include assisting in establishing test objectives and evaluation baselines, defining organizational responsibilities and relationships, and developing a reasonable schedule for testing. In addition, each member can contribute to preparing the request for proposal, and evaluating contractor proposals. The TPWG must be formed early to accomplish the above activities and to allow for test planning, and it remains in existence to update the TEMP and monitor test progress (source: AFR 80-14, August, 1980 para 19).

3 Computer Resources Working Group (CRWG) Meetings.

One of the tasks of the computer resource working group (CRWG) is to prepare and update the computer resources integrated support plan (CRISP) and ensure necessary elements of the CRISP are included in transfer and turnover agreements. The CRWG is initially chaired by the program office, and consists of members from the implementing, supporting, and using commands. Normally a draft CRISP will be circulated about six months after the CRWG first meets; after a few meetings to refine it, the CRISP is coordinated with the appropriate commands and approved by the program manager. The chairmanship of the CRWG is normally assumed by the

supporting command after PMRT (source: AFR 800-14, vol II, Sep 75, para 3-10). The CRWG will also review the OS/CMP, statements of work (SOW), and data item descriptions (DID) and recommend any contractor software support, to include outlining the scope and intent of any proposed software independent verification and validation.

There are a number of formal technical reviews of engineering efforts detailed in MIL-STD-499A and MIL-STD-1521, namely system requirements reviews (SRRs), system design reviews (SDRs), preliminary design reviews (PDRs), and critical design reviews (CDRs). Any of these reviews may actually involve a series of meetings. Of these types of reviews, the software test manager should certainly consider attending the last two.

4. Preliminary Design Review (PDR) Meetings.

A preliminary design review (PDR) should be conducted for each configuration item identified as part of the system to evaluate the progress, consistency, and technical adequacy of a selected design and test approach. From the software point of view, typically a PDR will review interfaces between computer program configuration items (CPCIs), implementation design of word lengths, message formats, available storage, timing and sizing data, and the test requirements, documentation, and tools (source: AFR 800-14, volume II, Sep 75, para 4-9).

5. Critical Design Review (CDR) Meetings.

A critical design review should be conducted on each configuration item to determine the acceptability of detail design requirements (Part I specifications), how these design requirements will be interpreted in the product specifications (Part II), performance, and test

characteristics as depicted in the specifications. For CPCIs the purpose is to establish the integrity of computer program requirements to the design at the level of flow charts prior to coding and testing. A CDR will typically include analyzing detailed flow charts, reviewing interactions with the data base, reviewing test plans and procedures for satisfying development specifications, and reviewing computer loading, iteration rates, processing time, and memory estimates (source: AFR 800-14, vol II, Sep 75, para 4-9).

6. Other Meetings.

Other meetings which may be of importance to the software test manager (depending on the agenda, etc.) are engineering design reviews (EDRs), progress reviews on IV&V or software, software audits (physical configuration audits (PCAs) or functional configuration audits (FCAs)), and, of course, any software test team meetings required. Other areas are a program's system design reviews (SDRs, normally at too high a level to help the software test manager except for general familiarization), source selection evaluations (normally not in AFTEC's purview), and periodic contractor to government briefings on progress, such as the program management review.

M. AFTEC PRODUCED PUBLICATIONS.

Besides a writeup of system deficiencies and strengths uncovered during testing, the most important tangible product to come out of AFTEC is the final OT&E report, forwarded to the Chief of Staff of the Air Force, with copies sent to interested parties. However, there are many test planning documents that must be written and implemented in order to create the testing whose results are described in the final report. The following paragraphs outline those publications to which the software test manager (and the TEBC advanced planner) will be contributing.

1. OT&E Approach.

A relatively short and intensive phase of AFTEC advance planning is conducted by the AFTEC program planning group (PPG) (chaired by the XRB program manager) in order to allow the AFTEC staff to scope the OT&E of a program in sufficient detail to support an estimate of the time and resources required by AFTEC to conduct the program. This phase is required to provide the system program manager these OT&E requirements early enough to incorporate them in his early program planning, especially for such long lead time items as simulations and test range equipment. The planning completed during this phase is structured into an OT&E approach briefing which is reviewed at the HQ AFTEC directorate level, thus providing an opportunity to approve or redirect the early planning efforts. The TEBC advanced planning manager should participate in this staff planning work by reviewing the briefing before it goes to TE to ensure that software concerns have been addressed. If a software test manager is appointed to the program this early, he will take over from the TEBC advanced planner in this respect.

2. OT&E Concept.

The OT&E concept consists of the latest refinements to the OT&E approach, together with the initial Test Program Outline (TPO) for the program. It serves as a transfer document. The XR program manager documents the program and OT&E planning in it prior to transfering the program to TE. After division level coordination, the concept is briefed to the directors, and eventually the commander, for his approval. The document is then distributed to the implementing and participating commands. Comments received back are considered in subsequent planning iterations.

The software test manager (or advanced planner) has the same responsibility here as with the OT&E approach--to review the briefing and ensure that software has been addressed adequately. Since the resource estimate, which is included as an attachment to the concept, will form the basis for the TPO, it is important that the

software test manager ensure that a deputy for software evaluation and a sufficient number of software evaluators have been included.

3. Test Plan (Software Annex).

After the test planning responsibility is transferred to TE, the test manager will convene a meeting of HQ AFTEC test team element members, the test director, and test team members (when available) to review the test design. Using and supporting commands should be solicited for inputs into the plan. The test manager will assemble inputs from the members into a draft test plan. The test plan will be forwarded to appropriate commands for working level reviews. After comments have been considered and incorporated, the final draft will be sent to all commands for coordination, as directed by AFTECR 55-1, section 4-4. The software test manager will ensure that software issues (if any) are appropriately addressed. There is a provision made for incorporation of a separate annex for software. Typically, especially for software intensive programs, this annex will be included and will contain objectives and evaluation measures for software issues not otherwise covered. Generally the format and contents are the same as for annex A (operational effectiveness) (see AFR 55-43 or AFTECR 55-1, Ch 4, attachment 1, p 16). Appendix 4 to this report is a sample outline for use. Note software specific items are included; e.g., software structure, responsibilities of software evaluators, etc. The appendix also contains checklists for items to be included in the software annex of the test plan.

4. Final Report.

The final report concentrates on presenting a clear picture of the results, conclusions, and recommendations derived from the OT&E. The first draft will be written at the test site, primarily by the test team. Succeeding drafts will be prepared at HQ AFTEC (using the Word Processing Center). The final draft of the test report will be submitted to TE for review. The final report must be reviewed by all directorates. The test manager is responsible for

coordinating the final report for AFTEC/CC approval. The software test manager is responsible for ensuring that software deficiencies and recommended corrective actions are appropriately documented and stressed.

5. Case File.

The test program case file contains the information and documentation (test plans/reports, messages, letters, data, etc.) necessary to maintain a record, from initiation to completion, of an AFTEC-managed/monitored test program.

AFTEC's involvement in a test program, whether managed or monitored, will normally determine the contents and detail of the case file and the amount of information and documentation that must be retained for historical purposes. Material contained in a case file that has legal, technical, and research value will be forwarded to the Washington National Records Center (WNRC) through HQ AFTEC for permanent retention. For this and the following reasons, it is important that a case file be maintained in an orderly fashion at all times.

a) During the conduct of a test program, the case file provides a ready reference for test program management purposes.

b) If AFTEC test program personnel are transferred, a complete and up-to-date case file greatly facilitates handover of the test program, prevents delays, and ensures that information is available to new personnel.

c) The case file shows the chronology of the test program and provides information to writers of the interim and final test reports.

d) After the test program is completed, the records maintained in the field that should be retained in the case file will be shipped to HQ AFTEC/DA.

Test directors and test managers/monitors must determine, by test program and the degree of AFTEC involvement, how much information and documentation they require in the case file.

The program case file plans are maintained by TEB for the software test managers. Contents are spelled out in chapter 10 of AFTECR 55-1. The software test manager will maintain his case files and, prior to step d) above, will consolidate his files with those of the test manager.

SECTION III

SOFTWARE TEST MANAGER LESSONS LEARNED

A. PERSONNEL LESSONS.

1. Deputy for Software Evaluation.

Certain characteristics are desirable in the choice for the deputy for software evaluation:

a) The deputy for software evaluation (DSE) should be brought on board early to assist in detailed software OT&E planning and to get familiar with the system.

b) It is imperative that the software test manager and the DSE have a good working relationship with each other, the contractor, and the program manager.

c) It is imperative that the DSE is a self-motivator. If not, test team motivation becomes a problem.

d) The DSE must be dedicated to the test for the entire test period, including final report writing.

e) The DSE should be an AFTEC resource of equal rank to the deputy for logistics and the deputy for operations.

2. Motivation.

Part of the duties of the software test manager include leadership responsibilities.

a) Don't let yourself or your software concerns get run over in the test planning group. Keep TEBC and TEB informed of any potential problem areas.

b) It is important to keep evaluators motivated. This can be difficult, particularly if there is a break in testing or program problems such as delays or money cut backs, etc.

3. Miscellaneous.

Performance Reports (OERs, APRs). All non-AFTEC evaluators on test teams still have strong ties to their parent command when it comes to loyalties, but they relate to their reporting official. This is another argument for having the deputy for software evaluation as an AFTEC slot.

B. EVALUATION LESSONS.

1. Preparation.

The software test manager (and the DSE) need all the documentation and source code listings in the latest format in time to perform evaluations. Promised deliveries are likely to cause problems unless the contractor (and the program office) are highly reliable.

The requirement for the use of the event trace monitor (ETM) must be identified early so that the system can be designed to accommodate the instrumentation.

2. Motivation.

The software test manager needs to motivate evaluators to perform the questionnaire evaluation. This can perhaps best be done with an inspiring pre-brief and dry run calibration.

Be aware that it is difficult to keep evaluators motivated if they are given too many programs to evaluate. As a rule of thumb, more than thirty questionnaires to answer is past the point of diminishing returns for evaluators.

When the software on a program is known to be highly unstable (such as is often the case early-on in an OT&E effort), be aware that the products provided and the enthusiasm towards evaluation are equally unstable. Such factors tend to degrade the resulting evaluation.

3. Contractor Maintained Software.

Software to be operated or maintained by contractors, especially if they are also the software's developers, pose special problems, to include:

a) Obtaining manning for the test team ("You don't need evaluators - this software is going to be contractor maintained!").

b) A natural tendency towards evaluating the contractor rather than the software.

c) The problem of evaluating non-deliverables to the Air Force.

The response? We evaluate the software's maintainability anyway to determine: the corporate committment to supporting Air Force needs, the responsiveness of the contractor to requests for software changes, etc. See lessons learned #5.

4. Software Failures.

The software test manager may well find himself embroiled in philosophical discussions with his test manager over terminology and definitions due to the unique nature of software. One example of this is the term "software failure." The argument can be made that there is no such animal as a software failure, in that software does not fail, it does exactly what it is programmed to do, and given the same set of conditions, it is exactly repeatable. Obviously hardware terms such as mean time between failure are not applicable to software in this sense. For the most part "software failures" should be renamed "software design errors," but in any event, it is imperative that the software test manager comes to an understanding with his test manager early in the test planning phase of the program.

5. Software OT&E Supporting Documentation.

During the period of active OT&E of a system, it is essential that the DSE maintain an active accounting of the progress of the

software evaluation. One method for keeping the OT&E test team informed of the progress of the software evaluation is by monthly submittal of memos for record or interim-reports to the software test manager and OT&E test director. These documents should be in the format of the OT&E test report (reference AFTEC OI-80-14) and detail each test objective and subjective and the appropriate results to date with supporting data. This format will facilitiate the preparation of the test report and keep all members of the OT&E headquarters and field test team informed of the status of the software evaluations.

6. Final Report Writing.

It is not uncommon for the test team to write the final report, striving for correct technical content, and then observe the Headquarters rewriting the report for format and content. Time constraints and "pride-of-authorship" feelings can increase tensions between the test team and the HQ element. One solution is to work together earlier and review final report drafts earlier.

Write the report for a wide spectrum of readers. With this in mind, keep "computerese" to the minimum, and aim the report for the least qualified reader.

7. Miscellaneous.

There needs to be a computer resources working group (CRWG) early in the program in order to air CRWG gripes officially. (Although required by AFR 800-14, sometimes program offices need reminding.)

Prior to test and reporting, a definition must be accepted as to what is to be reported as "undetermined" versus "unsatisfactory." Items that are not available for evaluation cannot be judged "unsatisfactory;" they are "undetermined."

The area of operational effectiveness is still a serious weakness for software.

It may be common to experience nonavailability of operational and/or support concepts. Therefore, use your own good judgment and talk to the users and supporters.

C. SPECIFIC LESSONS LEARNED.

The following pages contain specific lessons learned by software test managers on various programs. This section represents the continually changing data base of experience acquired by software test managers and is expected to grow with each new program managed by software branch personnel.

Lessons Learned #1

Program:

GPS User Equipment (UE)

Date:

18 Nov 80

Topic:

IV&V Contractor Support in OT&E Software Test Design

Lessons Learned:

For some acquisition and development programs of new weapon systems, an Independent Verification and Validation (IV&V) contractor is acquired by the program office to independently monitor and evaluate software associated with the weapon system. This resource (the IV&V contractor) can provide to AFTEC the expertise to identify potentially critical software paths and elements. Further, the IV&V contractor could assist AFTEC in designing IOT&E mission profiles to exercise or excite these potentially critical items, collect data, and analyze results. To obtain the services of the IV&V contractor, AFTEC must have the appropriate contract agreements.

Solution:

Since an IV&V contractor may be required during system acquisition, the desired method of obtaining support would be to include AFTEC specific tasks in the initial IV&V contract Statement of Work. Another approach would be to work with the program office to modify an existing contract with a change proposal. In either case, the tasks should be outlined as early as possible.

Key Words:

Software, Software Test Design, IV&V.

Lessons Learned #2

Program:

JTIDS

Date:

Dec 79

Topic:

Participation with Industry (PWI) Training

Lessons Learned:

JTIDS software training in preparation for the IOT&E took the form of PWI. While the training contract specified topics to be covered during the training, students received no formal classroom training nor an organized approach to cover the topics outlined in the contract. Rather, students were given free access to the technical library and were given opportunities to ask questions and to play with the system. The training, therefore, was not as effective as it should have been.

Solution:

When training takes the form of PWI, ensure the contract requires a minimum of 20% classroom training and a structured plan to lead the students through the material they should learn.

Key Words:

Software, Training, Participation with Industry.

Lessons Learned #3

Program:

JTIDS

Date:

Oct 80

Topic:

Evaluation of Fault Isolation Software

Lessons Learned:

During the Joint Tactical Information Distribution/Adaptable Surface Interface Terminal (JTIDS/ASIT) Initial Operational Test and Evaluation (IOT&E), the performance of the Fault Isolation Software (FIS) was not evaluated. The FIS requires the ASIT to be brought down to load FIS and to locate the failed unit. Since many critical resources were involved in the IOT&E, the fastest means of fixing faults was sought. Contractor personnel who were responsible for maintaining the system were able to fix faults faster if they did not use FIS. Therefore, FIS was not used during the IOT&E.

Solution:

Units which fail during critical tests should be reinserted into the system at a later date. Air Force operators should then use the system fault isolation capabilities to identify the failed unit and to gather data to support fault isolation subobjectives.

Key Words:

Software, Fault Isolation

Lessons Learned #4

Program:

GLCM

Date:

10 Nov 80

Topic:

Software Maintenance Concept

Lessons Learned:

To perform a software maintainability evaluation, it is necessary to know how and by whom the software will be maintained and what resources are being procured to support the concept. On the GLCM program, a number of the software subsystems are being considered common between Air Force and Navy applications (whether the systems will truly remain common once they become operational within each service is another issue, given the differences in philosophy and management of software practiced by each service). Due to this proposed commonality, the management and implementation of software changes and the overall configuration control of the GLCM software have yet to be determined. When an inadequate IV&V effort is added, we are presented with a dilemma: who will support a software maintainability evaluation when no support agency has been identified and what degree of evaluation should be performed when documentation and management status are not adequately identified?

Solution (recommended):

AFTEC needs to emphasize more strongly the need for a comprehensive software maintenance concept early during program development for all software being developed. If a concept is not known early in the planning of IOT&E, AFTEC should project the most pessimistic evaluation approach. For example, plan for some level of independent contractor support in assessing software maintainability. In this way, funding may force the issue or at least account for the lack of necessary information by directing a maintainability assessment and ignoring the cost-effectiveness factor.

Key Words:

Software, Maintainability

Lessons Learned #5

Program:

M-X

Date:

26 Nov 80

Topic:

Contractor Maintenance of Software

Lessons Learned:

Operational software for most space and missile systems is contractor maintained. Since there is no Air Force software support agency from which to draw test team people, the usual software suitability evaluation becomes infeasible and requires a different approach to be realistic. Some agencies even challenged the need and propriety of evaluating maintainability characteristics and contractor computer support resources. While the methodology must be modified, software maintainability must still be evaluated for the following reasons: (1) contractors typically bring in a different, lesser-qualified team once the software has been developed and it goes into a redevelopment or maintenance phase, (2) the software maintenance concept could change resulting in the Air Force having to assume maintenance responsibility, (3) the software development contractor could go out of business resulting in another contractor or agency having to assume maintenance responsibility, or (4) the contractor's performance may justify termination of his contract and awarding the software maintenance or redevelopment contract to an alternative contractor or agency.

Solution:

Always plan to evaluate software maintainability, but realistically tailor the methodology to the program. Suggestions follow: A system evaluation/technical direction (SE/TD) contractor or better yet, an independent verification and validation (IV&V) contractor to the program office could be used. Expand the scope of the SE/TD or IV&V contract to include evaluation of software maintainability using AFTEC questionnaires as a guide. If successive versions of software are expected to be developed during the test period, consider evaluating maintainability more directly. This can be done by tracking requirements for software changes through the process. Evaluate resources involved, complexity of change, and time to complete and verify each change. Other possibilities may be suggested by examining the specific software development and test process and agencies involved.

Key Words:

Software, Software Maintainability, Contractor Maintenance, Suitability Evaluation, Software Suitability.

Lessons Learned #6

Program:

Space Defense Systems Program

Date:

26 Nov 80

Topic:

Software Operational Effectiveness Evaluation

Lessons Learned:

Neither OT&E objectives nor delivery of test data for the software operational effectiveness evaluation was included in the independent verification and validation (IV&V) contract for the prototype miniature air-launched segment (PMALS). Because of extremely limited test resources, and because PMALS will be contractor operated and contractor maintained, IV&V is likely to be the only reasonable source of software effectiveness data.

Solution:

Scope the software operational test and evaluation concept and publish the concept or software OT&E test plan annex early. Distribute it widely. Emphasize the dependence on IV&V for evaluation data. Explore alternative sources but be reasonable and establish a credible need for data. List specific data requirements and when you need them to evaluate software effectiveness. Work with the program office to incorporate the requirements into the IV&V contract.

Key Words:

Software IV&V, Operational Effectiveness

Lessons Learned #7

Program:

F-16 Phase I FOT&E

Date:

20 Nov 80

Topic:

Software Trouble Reports

Lessons Learned:

For systems with extensive software, many of the operational problems encountered during testing arise from the computer program design and implementation. Some of these operational problems can be identified as design deficiencies and will be corrected by the contractor under corrections-of-deficiencies clauses of the contract. Many other problems are beyond the scope of the contract and will have to be corrected through Engineering Change Proposal (ECP) actions.

Current support concepts for embedded computer system software revolve around the use of a block change; that is, collecting a number of software change requirements for concurrent implementation in one development cycle. If even one change is to be developed, it is advantageous to make as many others concurrently to conserve development, test, documentation, and implementation resources (the overhead). As a result, the changes implemented in such a cycle are not usually restricted to corrections of deficiencies, but may include capability improvements or even enhancements. However, seldom are there enough resources, or is it technically reasonable, to attempt to implement all reported change requests in one block change cycle. Consequently, a prioritization of all software trouble reports (change requests) are necessary to select the specific changes for development in that cycle.

To support the process of defining the specific changes to be implemented in a given block change cycle, a single source document containing all the software change candidates is needed. Such an "Operational Software Requirements Document" (OSRD) should be established at the beginning of OT&E testing when the first production baseline software is available. Software trouble reports should be added to the OSRD as they are identified, and the OSRD containing all software change candidates should be provided as the major input to the design requirements process for the first and all subsequent block change designs.

Solution:

The Deputy for Software Evaluation or other members of the OT&E test team should ensure that a system is used for identifying and tracking all test results that may cause changes to an embedded computer program. These changes include corrections to design deficiencies, system operational characteristics which do not fit operational tactics, procedures, or other considerations, and system improvements or enhancements which are feasible within the constraints imposed by the system hardware (i.e, which may be implemented through a software-only change).

This identification and tracking system should be integrated with the service report processing system (IAW TO 00-35D-54, AFR 55-43, and test team operating instructions) but will typically require additional procedures at the review stage to adequately assess the impact of the change request, in the tracking activities to ensure that reports are consolidated into a single document, and in the action stage to coordinate and integrate all change requests into a cohesive set for defining requirements for a block change cycle.

Key Words:

Software, Block Changes, Change Requests, Deputy for Software Evaluation, Service Reports

Lessons Learned #8

Program:

ALCM

Date:

10 Nov 80

Topic:

Software Documentation

Lessons Learned:

During the ALCM competition we depended on the Joint Cruise Missile Program Office (JCMPO) to ensure the availability of adequate software documentation to perform a software maintenance assessment during IOT&E. We were limited in our direct contact with the development contractors due to the JCMPO approach to the handling of the contracts. Because of the JCMPO serving as a moderator between AFTEC and the contractor, documentation was delayed and quantities were insufficient. If the competition had not slipped, the software maintainability evaluation would not have been completed. The basic problem was JCMPO delays and incorrect presentation of our requirements to the contractors.

Solution (recommended):

Once we were allowed to discuss the AFTEC requirements directly with the contractors, the situation improved. It is imperative that AFTEC brief development contractors as to the evaluation techniques to be used for software evaluation and the data requirements necessary to support this evaluation. This briefing should take place early in the development, preferably prior to preliminary design review, so that any impact caused by our requirements can be identified early. Once direct contact is made between AFTEC and the contractor, it must continue throughout the evaluation phase. This requirement applies to all types of program acquisition, particularly competitions where the scheduling of resources and documentation deliveries are more difficult.

Key Words:

Software, Software Maintainability

SECTION IV

SOFTWARE TEST MANAGER'S HANDBOOK
ACRONYMS AND GLOSSARY

A. ACRONYMS.

A table of acronyms that should prove useful to the software test manager is provided on the following pages.

ACRONYMS

ADCOM	Air Defense Command
ADP	automated data processing
ADPAG	AFTEC ADP advisory group
ADPE	ADP equipment
AFFTC	Air Force Flight Test Center
AFLC	Air Force Logistics Command
AFM	Air Force manual
AFR	Air Force regulation
AFSC	Air Force specialty code
AFSC	Air Force Systems Command
AFTEC	Air Force Test and Evaluation Center
AFTO	Air Force technical order
AGL	above ground level
AGM	air-to-ground missile
AIS	avionics intermediate shop
AISF	avionics intermediate support facility
ALCM	air-launched cruise missile
ALC	air logistics center
APL	a programming language
APR	Airman Performance Report
ASD	Acquistion Systems Dividion (AFSC)
ASIT	adaptable surface interface terminals (for JTIDS)
ATD	aircrew training device
ATC	Air Training Command
ATE	automatic test equipment
ATEC	automated technical control
ATLAS	abbreviated test language for all systems (ATE language)
AV	air vehicle
AVE	air vehicle equipment
AVI	air vehicle inventory
BIT	built-in test
CDR	critical design review
CDRL	contract data requirements list
CEP	circular error probable
CG	center of gravity
COBOL	common business-oriented language
COMOPTEVFOR	Commander, OPTEVFOR
CND	could not duplicate
CPCI	computer program configuration item
CPDP	computer program development plan
CPOR	computer program observation report
CRISP	computer resources integrated support plan
CRWG	computer resources working group
CTF	Combined Test Force
DART	deficiency analysis review team
DCP	decision coordinating paper

ACRONYMS (continued)

DID	data item description
DMA	Defense Mapping Agency
DMP	Data Management Plan
DOD	Department of Defense
DOE	Department of Energy
DR	discrepancy report (replaced by SR)
DSARC	Defense System Acquisition Review Council
DSE	Deputy for Software Evaluation
DT&E	development test and evaluation
DTD	data transport device
EAFB	Edwards Air Force Base
EAROM	electrically alterable ROM
ECP	engineering change proposal
ECS	embedded computer system
EMC	electromagnetic compatibility
EMI	electromagnetic interference
EPROM	ultraviolet eraseable ROM
ESD	Electronic System Division (AFSC)
ESTS	electronic system test set
ETI	elapsed time indicator
ETM	event trace monitor
EWO	emergency war order; electronic warfare office
FAD	Force Activity Designator
FDI	fault detection isolation
FIT	fault isolation test
FMC	full mission capable
FORTRAN	formula translation
FOT&E	follow-on operational test and evaluation
FSD	full-scale development
FSED	full-scale engineering development
FTU	flight test unit
GFE	government-furnished equipment
GLCM	ground-launched cruise missile
GPS	global positioning system (or NAVSTAR GPS)
GSERD	ground support equipment recommendation data
HARM	high-speed anti-radiation missile
HF	human factors
IAW	in accordance with
ICS	interim contractor support
ILS	integrated logistics support
IMF	integrated maintenance facility
INS	inertial navigation system
IOC	initial operational capability
IOT&E	initial operational test and evaluation
IV&V	independent verification and validation

ACRONYMS (continued)

JCMPO	Joint Cruise Missiles Project Office
JOVIAL	Jules' own version of the international algebraic language
JPO	joint project office
JRMET	joint reliability maintainability evaluation team
JSTPS	joint strategic target planning staff
JTIDS	joint tactical information distribution system
JTU	joint test unit
LOE	letter of evaluation
LRU	line replaceable unit
LSA	logistics support analysis
LSAR	logistics support analysis records
LSET	logistics suitability evaluation team (no longer in vogue)
MAJCOM	major command
MCSP	mission completion success probability
M DEMO	maintainability demonstration
MDPS	mission data preparation system
MENA	mission element need analysis
MENS	mission element needs statement
MIL-STD	military standard
MIP	materiel improvement project
MMH	maintenance man-hours
MMS	munition maintenance squadron
MOE	measure of effectiveness
MQT	military qualification-test
MRB	materiel review board
MSL	mean sea level
MTBCF	mean time between critical failure
MTBD	mean time between demand
MTBF	mean time between failure
MTBMa	mean time between maintenance action
MTT	maintainability task time
MTTR	mean time to repair
NDI	nondestructive inspection
NMC	not mission capable
NRTS	not repairable this station
OAS	offensive avionics system
OC	operating cycle
OC-ALC	Oklahoma City Air Logistics Center
ODDL	onboard digital data load
OER	Officer Effectiveness Report
OFP	operational flight program
OFS	operational flight software
OH	operating hour
OI	office of information
OL	(AFTEC) operating location
OO-ALC	Ogden Air Logistics Center

ACRONYMS (continued)

OPEVAL	operational evaluation
OPTEVFOR	Operational Test and Evaluation Force (Navy)
ORLA	optimum repair level analysis
O&M	operation and maintenance
O&S	operations and support
O/S CMP	Operational/Support Configuration Management Procedures
OSE	operational support evaluation
OT&E	operational test and evaluation
OTEA	Operation Test and Evaluation Agency (Army)
PDR	preliminary design review
PLSS	precision location strike system
PMC	partial mission capable
PMD	program management directive
PME	precision measuring equipment
PMEL	precision measurement equipment laboratory
PMP	program management plan
PMR	Pacific Missile Range
PMRT	program management responsibility transfer
PMTC	program manager's training course; point magu training center
PROM	programmable read only memory
PRVT	product reliability verification test
QAP	questionnaire analysis program
QOT&E	qualification operational test and evaluation
QPA	quantity per aircraft
RCC	remote command and control
R&M	reliability and maintainability
ROC	required operational capability (replaced by SON)
ROM	read only memory
RTO	responsible test organization
SA-ALC	San Antonio Air Logistics Center
SAC	Strategic Air Command
SAF	Secretary of the Air Force
SAMSO	Space and Missile Systems Organization (replaced by SD)
SAMTEC	Space and Missile Test Center (replaced by SD)
SAT	software assessment team (replaced by DSE)
SD	Space Division
SE	support equipment
SEI	support equipment inventory
SERD	support equipment recommendation data
SEW	support evaluation worksheet
SI	special inventory
SIOP	single integrated operational plan
SLCM	sea-launched cruise missile
SM	system manager
SM-ALC	Sacramento Air Logistics Center
SMAP	software maintainability analysis program

ACRONYMS (continued)

SOMIQ	software operator-machine interface questionnaire
SON	statement of operational need
SPO	system program office
SPR	software problem report
SR	service report
SRAM	short-range attack missile
SRU	shop replaceable unit
SSF	software support facility
SSPO	Strategic System Program Office
S/W	software
TAC	Tactical Air Command
TAF	Tactical Air Forces
TAWC	Tactical Air Warfare Center
TBD	to be determined
TCTO	time compliance technical order
TD	test discrepancy; test directive; test director
TDY	temporary duty
TEMP	Test and Evaluation Master Plan
TEP	test and evaluation plan
TERCOM	terrain contour matching
TIP	test and integration plan
TIS	test information sheet
TO	technical order
TOA	time of arrival
TOCU	technical order control unit
TOMA	technical order management activity
TOT	time over target
TP&H	transportation, packaging and handling
TPO	test program outline
TPR	trained personnel requirement
TRC	technical repair center
TSE	training supportability evaluator
TSPI	time-space-position information
TSTM	training supportability test manager
UDL	unit detail listing
UOT	user oriented testing
USDR&E	Under Secretary of Defense for Research and Engineering
UTTR	Utah Test and Training Range
UUT	unit under test
WR-ALC	Warner-Robins Air Logistics Center
WRM	war reserve material
WUC	work unit code

B. GLOSSARY.

The following selected definitions should prove helpful to the Software Test Manager as a reference guide. Sources indicated include the following:

1) DOD Directive 5000.3 (encl 1).
2) AFR 80-14 (attachment 1).

1. Critical Issues.

Those aspects of a system's capability, either operational, technical, or other, that must be questioned before a system's overall worth can be estimated, and that are of primary importance to the decision authority in reaching a decision to allow the system to advance into the next acquisition phase. (DOD Directive 5000.3)

2. Embedded Computer System.

A computer that is integral to an electro-mechanical system, and that has the following key attributes:

a) Physically incorporated into a large system whose primary function is not data processing.
b) Integral to, or supportive of, a larger system from a design, procurement, and operations viewpoint.
c) Inputs target data, environmental data, command and control, etc.
d) Outputs include target information, flight information, control signals, etc.

In general, an embedded computer system (ECS) is developed, acquired, and operated under decentralized management. (DOD Directives 5000.1, 5000.2.)

3. Software.

A set of computer programs, procedures, and associated documentation concerned with the operation of a data processing system.

NOTE: Software Intensive means computer applications where the functions are dynamic, likely to change, and where the ability to change the functions is considered an asset.

4. Firmware.

Defn. a: Computer programs and data loaded in a class of memory that cannot be dynamically modified by the computer during processing.

Defn. b: Hardware that contains a computer program and data that cannot be changed in its user environment.

NOTE 1. The computer programs and data contained in firmware are classified as software; the circuitry containing the computer program and data is classified as hardware. (Data and Analysis Center for Software).

NOTE 2. Hardware Intensive means computer applications in which the function is fixed and hence, the computer program, after development and test, is not expected to be changed for the lifetime of the physical component in which it is embedded.

5. Software Maintainability.

A measure of the ease with which software can be changed in order to:

a) Correct errors.
b) Add or modify system capabilities through software changes.
c) Delete features from programs.
d) Modify software to be compatible with hardware changes.

6. Evaluation Criteria.

Standards by which achievement of required operational effectiveness/suitability characteristics, or resolution of technical or operational issues may be judged. At milestone II and beyond,

evaluation criteria must include quantitative goals (the desired value) and thresholds (the value beyond which the characteristic is unsatisfactory) whenever possible. (DOD Directive 5000.3)

Under Air Force Manual 55-43, evaluation criteria consist of goals, standards, and thresholds.

7. Independent Verification and Validation. (IV&V)

An independent software assessment process structured to ensure that the computer program fulfills the requirements stated in system and subsystem specifications and satisfactorily performs, in the operational environment, the functions required to meet the user's and supporter's needs. IV&V consists of three essential elements - independence, verification, and validation:

Independent: An organization/agency which is separate from the software development contractor(s) from a contractual and organizational standpoint.

Verification: The iterative evaluation to determine whether the products of each step of the software development process fulfills all requirements levied by the previous step.

Validation: The integration, testing, and evaluation activities carried out at the system/subsystem level to ensure that the finally developed computer program satisfies the system specification and the user's/supporter's requirements. (AFR 80-14)

8. Operational Effectiveness.

The overall degree of mission accomplishment of a system used by representative personnel in the context of the organization, doctrine, tactics, threat (including countermeasures and nuclear threats), and environment in the planned operational employment of the system. (DOD Directive 5000.3)

9. Operational Suitability.

The degree to which a system can be satisfactorily placed in field use, with consideration being given availability, compatibility, transportability, interoperability, reliability, wartime usage rates, maintainability, safety, human factors, manpower supportability, logistic supportability, and training requirements. (DOD Directive 5000.3)

APPENDIX 1

PMD CHECKLIST - AFTEC MANAGED PROGRAM[3]

(Source: HQ AFTEC/XR)

1. GENERAL SUGGESTION: Draft or published PMDs come in many shapes, from "terrible" to "excellent". If you're working with a "terrible" one, you should consider referring to HQ USAF OI (HOI) 800-2, atch 2, for detailed guidance. Copies are available in XRB, Room 113 for reference, and additional copies will be distributed for the AFTEC staff upon publication (anticipated in Jan-Feb 80). Also, XRB is developing a file of examples of "good" PMDs for your future reference. Points of contact are Major Irwin or Major Howell, ext. 4-5792.

2. Para 1, SPECIFIC PURPOSE. Should read: "AFTEC is designated the Operational Test and Evaluation Agency;" or (alternative format): "OT&E command/agency - AFTEC."

3. Para 2, PROGRAM SUMMARY. For Multiservice Programs ensure appropriate words are included to designate the lead service for the system and the lead OT&E agency for testing. For Multinational programs, ensure that any relevant govt - govt MOUs are referenced, and that a DOD lead service is designated (if appropriate).

4. Para 3, PROGRAM MANAGEMENT DIRECTION[2]. Include/add the following subparagraphs.

"a. Intelligence/Threat Estimate[1]:

"(1) The threat data required by this PMD will be provided in accordance with AFR 800-3 by (the implementing command or AFSC) from a current Threat Environmental Description (TED). (The implementing command or AFSC) will update or develop the TED and obtain AF/IN approval by (one year from date of this PMD or month/year). If a current and appropriate TED exists, approval for its use will be obtained from AF/IN. The TED will contain or reference sufficient threat data to accomplish interactive analysis, per DODD 5000.2, for system engineering survivability/vulnerability analysis (AFR 80-38), threat simulation for test and evaluation (AFR 80-14), security decisions and technology exploitation. From the interactive analysis, the threat parameters and issues/concerns (critical intelligence parameters) will be defined and forwarded by (the implementing command or AFSC) to the operating, participating, OT&E, and supporting commands and other authority for comment. (The TED can provide the threat data required by a Threat Working Group). The timeliness and detail of the OT&E command's threat data requirements will be considered in preparing the TED.

"(2) Approximately one year prior to each scheduled DSARC or equivalent milestone review, (the implementing command or AFSC) will initiate the preparation of a Threat Assessment Report (TAR) for AF/IN approval and processing. The TAR, an updated extract from the TED report, will emphasize specific system features and critical intelligence parameters. In order to treat the system in the broadest sense, the TAR will encompass the threat for the projected life of the system and may contain data concerning the target system."

(NOTE: Since these words are rather lengthy, a shorter way to get the point across would be: "P.3, para 3a(1), Subj: Threat Environment Document. Replace the current subpara with wording approved by HQ USAF/INE".)

b. System Operational Concept/Suspense[1]. "XXX (lead MAJCOM) will prepare and maintain a current system operational concept IAW AFR 57-1 (including a maintenance concept IAW AFR 66-14) and forward to HQ USAF/XOO/RD_/LEY for review and approval NLT (date desired)."

c. TEMP/SUSPENSE[1]. "AFSC will prepare a Test and Evaluation Master Plan (TEMP) with OT&E inputs provided by AFTEC, and forward to HQ USAF/RD_/XOO for review NLT (date)."

d. AFTEC. "AFTEC will manage (system) IOT&E (or QOT&E) IAW AFR 80-14." If the decision has not been made whether AFTEC or the operating command will conduct the OT&E, the PMD should direct AFTEC to "review the program, in conjunction with the operating and implementing commands, and make an OT&E management recommendation to HQ USAF/XOORE _____ for inclusion in a subsequent amendment of the PMD."

e. Address the requirement for an integrated logistics support (ILS) program IAW AFR 800-8.

f. Address the requirement for a reliability and maintainability program IAW AFR 80-5.

g. Address the requirement to analyze and consider alternative support concepts, with regard to the least LCC and support costs which meet mission requirements.

h. Para 3. Address the requirement for a computer resources integrated support plan (CRISP) IAW AFR 800-14.

i. Para 3. Address the requirement for an independent validation and verification (IV&V) of system software.

j. Para 3. Address the requirement for the development and acquisition of support equipment IAW AFR 800-12.

5. Para 4, Program Management Guidance.[2]

a. Schedules. Those significant schedules, milestones, briefing reviews and test dates for each phase should be addressed (if feasible).

b. The cost, schedule and performance goals, with associated constraints or thresholds, should be addressed (required for FSED).

c. Are Life Cycle Costs (LCC) addressed as a basis for decisions IAW the JLC Design-to-Cost Guide, and AFRs 800-3 & 800-11?

d. Address the development & acquisition of support equipment IAW AFR 800-12.

e. Test & Evaluation subparagraph[2,3].

(1) Address those critical questions, areas of risk & specific test objectives which HQ USAF should direct (including suitability, software, human factors, survivability and safety).

(2) Is the PMD direction in compliance with DODD 5000.3 and AFR 80-14? If not, a waiver should be documented.

(3) Ensure the responsibilities of the operating, supporting, training and implementing commands are clearly defined to provide adequate resources for OT&E (including suitability).

(4) Are test article & test support resources required for DT&E/OT&E identified? If not, are AFSC, AFTEC and other test commands/agencies tasked to identify requirements for inclusion in a subsequent PMD?

(5) Address the requirements for combined and/or separate DT&E/OT&E.

(6) For multinational/multiservice programs, address any special direction for participation with the lead service, preparation of joint services or govt-to-govt agreements on testing.

f. If adequate ECM/ECCM testing/simulation equipment is not available, direct the implementing command to develop or procure such equipment.

g. Address the requirement for frequency planning IAW AFR 100-31.

h. If manpower requirements for OT&E have previously been approved through manpower channels, include a statement if additional manpower is to be programmed by HQ USAF.

i. Address training equipment requirements including aircrew training devices.

6. Distribution Page. Should always read:

"AFTEC/XR. . . . . .5 copies."

NOTE 1: Items apply to major/AFDAP/designated nonmajor acquisition or modification programs, and only as guidance for lesser programs (see AFRs 57-1 and 66-14).
NOTE 2: Items may be addressed in either para 3 or 4 of the PMD, depending on whether Air Staff decides these items are to be directives or guidance only.
NOTE 3: If the decision has not been made whether AFTEC or the operating command will conduct the OT&E, the PMD should directed AFTEC to review the program and make an OT&E management recommendation for inclusion in a subsequent PMD (see para 4d of checklist).

APPENDIX 2

COMPUTER RESOURCES INTEGRATED SUPPORT PLAN FORMAT

(Source: AFLCR 800-21)

(A CRISP table of contents may be obtained from HQ AFLC/LOEC.)

1.0 Introduction.

1.1 OVERVIEW.

(Give the purpose of the CRISP and identify the system/subsystem it addresses. Include a brief program summary along with the structure of CRISP; i.e., number of volumes and subjects.)

1.2 APPLICABILITY.

(State any pertinent background not included in paragraph 1.1. Document the scope of the CRISP and the authority (i.e., PMD, AFR 800-14, etc.))

1.3 REFERENCES.

(This paragraph may include abbreviations/acronyms, glossary, and list of applicable documents. Basically relate the CRISP to other CRISP interfaces.)

NOTE: Attachment 1 explains several terms used in this sample.

1.4 SYSTEM DESCRIPTION.

(A brief description of this weapon system and/or subsystem and present status of the system. No a detailed description.)

1.5 PROCESSOR(S) AND SOFTWARE DESCRIPTION.

(Processor identification and a description of the software (firmware) associated with each system/subsystem. Microprocessor (attachment 1) applications be identified as software intensive or hardware intensive (attachment 1). Firmware should be classified as software intensive (SWIF) or hardware intensive (HWIF). A block diagraph should be included to provide graphic representation of the system.)

2.0 Management Approach.

2.1 MANAGEMENT FOCAL POINTS.

(This section should contain information on CRWG members, organizations involved, offices of primary responsibility (OPR) and their responsibilities, organizational structure, and interface description for pre- and post- PMRT. An organizational chart could be included.)

2.2 SUPPORT CONCEPT.

(Details of the support concept should include plans/procedures to establish and operate the support facility with reference to the management impacts. Phase charts for implementation may be included. Identification of funding requirements should be documented for all phases of life cycle support. Emphasis should be given to the tasks AFLC performs in support of development before PMRT for SPO budgeting; for instance, those tasks performed by the industrially funded Software Support Centers.)

2.3 SYSTEM/SUBSYSTEM TURNOVER.

(This section details the plans/procedures for operational and support system turnover. It gives procedures/plans for the support of computer programs during turnover.)

2.4 PROGRAM MANAGEMENT RESPONSIBILITY TRANSFER (PMRT).

(Paragraph gives procedures/plans for operational system before and at PMRT. Include the procedures/plans for support system operations pre- and post- PMRT. Separate PMRT plans may be referenced where appropriate, and will become a part of this plan to the extent at which they apply to computer resources.)

2.5 SOFTWARE CHANGES

(This paragraph details methodology and time constraints for reprogramming actions on the operational flight program (i.e., block changes, emergency changes.) Information should include pre- and post- PMRT.)

2.6 MODIFICATIONS

(Identify procedures for modifications to the system according to AFR 57-4.)

2.7 DEFICIENCY REPORTING

(Identify method for reporting deficiencies in computer programs and procedures to correct these deficiencies.)

3. Configuration Management.

3.1 GENERAL.

(Identify basic concepts for maintaining configuration control of the computer resources. Include references to applicable documents as appropriate. Also reference the appendix for designating the CPCI listing.)

3.2 CONFIGURATION CONTROL RESPONSIBILITIES

(This section should detail the change control authority, organizational responsibilities, and interaction interface between acquiring, using, and supporting commands. The information should cover pre- and post- PMRT.)

3.3 CHANGE CONTROL.

(Identify the plans and procedures for recommending, approving, and processing changes to the computer programs. These changes may be software only, software/hardware, and routine vs emergency change requirements. These plans and procedures should cover pre- and post-PMRT.)

3.4 STATUS ACCOUNTING.

(Identify the CPCI configuration baseline and procedures for accounting for implementation of the change(s).)

3.5 COMPUTER PROGRAM CONFIGURED ITEM IDENTIFICATION (CPIN)

(Identification of the computer programs as configured

items and procedures for assigning Computer Program Identification Number. Reference CPIN Compendium 80-1 and AFR 800-21, chapter 11.)

4.0 Documentation.

4.1 OPERATIONAL SYSTEM DOCUMENTATION.

(Identify the documentation required to support the operational system (i.e., organic, contractor) which will assume timely support of all involved computer programs according to the support concept. Include need dates and transfer methods.)

4.2 SUPPORT SYSTEM DOCUMENTATION.

(Identify support system documentation requirements for proper operation and support of involved computer resources. Include need dates and transfer methods.)

4.3 DOCUMENTATION CONTROL PLANS/ PROCEDURES.

(This section should cover the documentation control plans/ procedures for controlling and updating documentation. Also addresss storage plans/procedures for documentation.)

4.4 DOCUMENTATION INDENTIFICATION (CPIN).

5.0 Personnel and Training.

5.1 PERSONNEL.

(Identify personnel and specialty requirements for managing and supporting the computer resources involved. This section should also identify the contractor resources required for interim contractor support and funding responsibility.)

5.2 TRAINING.

(This section should identify the training required (formal and informal) to support the computer resources involved and ensure successful operation and management of the system.)

6.0 Support Equipment/Software and Facility Requirements.

6.1 EQUIPMENT REQUIREMENTS.

(Identify supporting command equipment required to support the operational software programs following PMRT. The concept for acquisition, integration, and operation of the support equipment and plans for verification and validation of the support equipment should be identified.)

6.2 SOFTWARE FIRMWARE.

(Identify the software firmware programs required, method for acquisition, integration, and operation, plans for verification, validation, and engineering analysis of the operational software; and related mission equipment interfaces.)

6.3 SUPPORT SOFTWARE (GENERAL).

(Describe support software (i.e computer, translators, scientific subroutines, media to media conversion programs, etc) required and identify associated documentation if not identified in Section 4.0.)

6.4 FACILITIES.

(Describe the requirements for physical housing of support equipment and plans to establish housing.)

6.5 MAINTENANCE OF COMPUTER RESOURCES EQUIPMENT.

(Identify a plan to maintain the equipment to include funding responsibilities.)

7.0 Test Support.

7.1 OPERATIONAL TEST AND EVALUATION.

(Identify field requirements, adaptation for operation personnel, special support equipment, special support procedures, interfacing agencies, and any special maintenance requirements.)

7.2 FLIGHT TEST.

(Identify aircraft requirements, flight instrumentation needed, test ranges to be used, and any special maintenance requirements.)

8.0 Verification and Validation (V&V).

8.1 OPERATIONAL SOFTWARE.

(Identify verification and validation and acceptance testing requirements for computer programs involved. Include plan and requirement for independent V&V of the operational software and any interfacing agencies. Responsibilities and procedures for V&V prior to and following PMRT.)

8.2 SUPPORT SOFTWARE.

(Identify verification and validation and acceptance testing requirements of support software. Plans and requirements for independent V&V of support software. Separate test plans and procedures developed for the purpose may be referenced. Responsibilities and procedures for V&V prior to and following PMRT.)

9.0 Security.

(Identify any special security handling procedures and the impact of the security procedures on operational support.)

10.0 Security Assistance.

(Identify the sale or possible sale of the system to a foreign country, the equipment/software that is not releasible to foreign countries, and the support concept/responsibilities.)

APPENDIX 3

COMPUTER PROGRAM OBSERVATION REPORT

COMPUTER PROGRAM OBSERVATION REPORT			SERIAL NUMBER 0010
I. COMPUTER PROGRAM IDENTIFICATION			
SYSTEM (A-7, B-52G, 487L): FB-111A	SUB-SYSTEM/COMPUTER (Nav system, Fire Control Computer): NCU	COMPUTER PROGRAM (OFP, Aerodynamic Data Update Program): OFP	SUB-PROGRAM MODULE (Include version designation): FB-13 Roll Rate

II. ORIGINATOR IDENTIFICATION (Investigator, coordinator, approving officials, etc)

NAME (Last, First, Middle initial)	GRADE	ORGANIZATION AND STATION	DUTY PHONE
EVALUATOR, I.M.	CAPT	AFTEC OL-CC	X 7432
DEPUTIE, U.R.	MAJ	AFTEC OL-CC	X 2198

III. OTHER DATA

DATE AND TIME OF OBSERVATION (If important, include elapsed times)	DEFICIENCY REPORT WATCH ITEM
14 FEB 81 1030 - 1130	[X] YES [ ] NO

DOCUMENT REFERENCES

DOCUMENT NO.	DATE	PUBLISHER (USAF, Rand Corp, etc)	PAGE NO.	FIGURE	SECTION	PARA NO.
UNK						

IV. OBSERVATION (Include what did/did not happen, what should have happened, results, suggested changes, relevant conditions, etc)

When the aircraft was in a vertical dive, the roll rate indicated the aircraft was spinning, when in fact it was not.

SAMPLE

AFTEC FORM 207 AUG 78

APPENDIX 4

FORMAT AND CHECKLIST FOR SOFTWARE ANNEX

EXAMPLE OF ANNEX D

_____ SOFTWARE OPERATIONAL TEST AND EVALUATION

D.1 INTRODUCTION.

This subsection should provide general descriptive information concerning the software test and evaluation. Try to keep it brief. This is a good place to present an overview (roadmap) of the software evaluation. Suggest this overview be presented in the format shown in figure 1.

D.1.1 Participating Organizations.

Discuss the composition of the software analysts/evaluators on the test team. Reference the TPO and list the number of personnel each organization has signed up to provide.

D.1.2 Responsibilities.

The role of the software analysts/evaluators and the deputy for software evaluation should be discussed in this subsection. You may wish to break out responsibilities under three headings as follows:

a) Deputy for Software Evaluation (DSE). Focal point for all software mat s.

b) Evaluators (PCS). In addition to completing questionnaires, assists the deputy for software evaluation in planning and evaluating other objectives.

Evaluators (TDY). Primarily responsible for completing questionnaires.

B-52 OAS SOFTWARE EVALUATION STRUCTURE

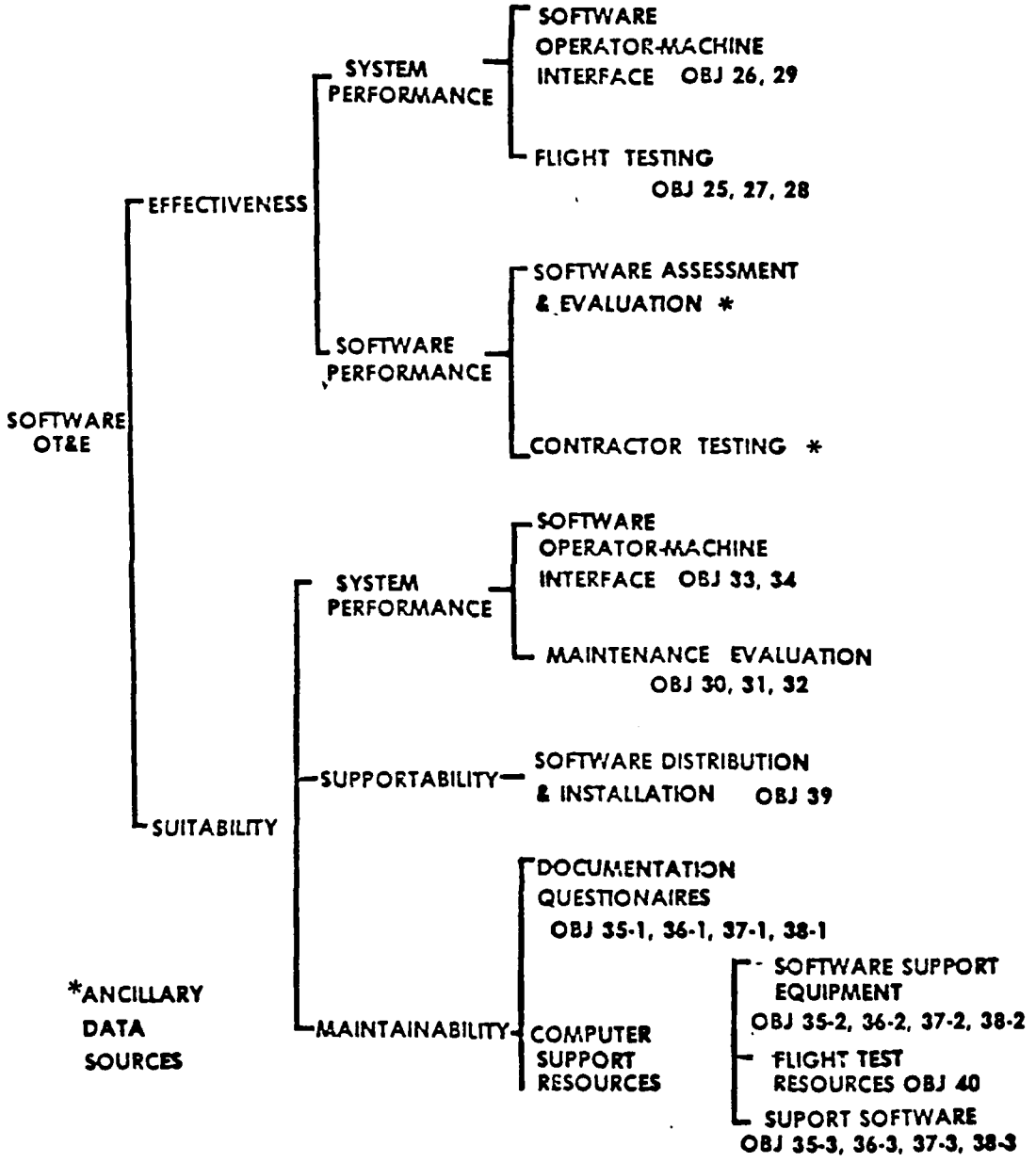

Figure 1. Example of Software OT&E Overview

Note: A strawman checklist of responsibilities for each heading is provided in attachment 1 to this appendix.

D.1.3 Evaluation Limitations.

Factors which limit the software evaluation should be discussed in this subsection.

D.1.4 Others.

Other factors you feel are important to discuss should be presented in this and subsequent subsections.

D.2 SOFTWARE DESCRIPTIONS.

This section should include a brief description of the functional operation of the software and identification of the major software programs being tested. Characteristics of the software such as language, structure, etc., should be identified. The software maintenance concept should also be discussed in this section.

D.3 SOFTWARE OPERATIONAL EFFECTIVENESS EVALUATION.

Types of software objectives that could be included as part of an operational effectiveness evaluation are presented in attachment 2 to this appendix. The format for the discussion of each objective is the same as that presented in annex A to the test plan and as shown below.

D.3.1 Objective.

D.3.1.1 Measures of Effectiveness/Evaluation Criteria.

D.3.1.2 Methodology.

D.3.1.3 Data Management.

D.3.1.3.1 Data Requirements.

D.3.1.3.2 Data Collection and Processing.

D.3.1.3.3 Data Analysis.

D.3.1.4 Evaluation.

D.4 SOFTWARE SUITABILITY EVALUATION.

Types of objectives that could be included as part of a operational suitability evaluation are presented in attachment 3 to this appendix.

ATTACHMENT 1

APPENDIX 4

RESPONSIBILITIES FOR SOFTWARE PERSONNEL SUPPORTING AN OT&E

A1.1 Responsibilities of Software Evaluators.

Under the guidance of the deputy for software evaluation, the evaluators will be responsible for making a unified assessment of the software. The responsibilities of software personnel supporting the OT&E are presented below.

A1.2 Responsibilities of the Deputy for Software Evaluation.

The focal point for all software evaluation matters will be the deputy for software evaluation. Specifically the deputy will:

a) Manage the software evaluators. This includes planning, scheduling, and coordinating activities and assigning evaluators to perform required functions.

b) Establish any unique procedures required for effective control of software related activities.

c) Coordinate software activities with other test activities and identify potential schedule or resource conflicts to the IOT&E test director for resolution.

d) Prepare and submit status reports, as required, to the test director.

e) Participate in the software configuration control process. Maintain cognizance of all software changes proposed and in various stages of implementation. Chair a software problem review board during OT&E.

A1.3 Responsibilities for Software Evaluators.

The software evaluators will be responsible for the following:

a) Complete software documentation and software source listing questionnaires.

b) Prepare Computer Program Observation Reports (AFTEC Form 207) to document anomalies or problems noted during the software suitability evaluation.

c) Assist the deputy for software evaluation in collecting, monitoring, and reviewing data for evaluating computer support resources.

A1.4 Responsibilities for Software Analysts.

In addition to the above software evaluators' responsibilities, the software analysts (PCS evaluators) will also be responsible for the following:

a) Collect, monitor, and review data for all software objectives.

b) Identify software discrepancies and monitor corrective actions.

c) Assist the deputy for software evaluation in administering the Software Operator-Machine Interface Questionnaires.

d) Assist the deputy for software evaluation in selecting software documentation and source listing to be evaluated.

e) Assist the deputy for software evaluation in preparation of the software assessment portions of the final report.

ATTACHMENT 2

APPENDIX 4

SOFTWARE OBJECTIVES TO SUPPORT OPERATIONAL EFFECTIVENESS EVALUATIONS

A2.1 Effectiveness Subobjectives.

The operational effectiveness evaluation for software objectives typically will be addressed under the categories of performance and operator-machine interface of the mission applications software as shown in figure 1. The performance objective can be divided into two types of subobjectives: system performance subobjectives and software peculiar performance subobjectives.

A2.2 Methods for Formulating Performance Subobjectives.

There are three methods the software test manager can use to arrive at performance subobjectives:

a) Method 1. Define the functions the software must perform for the system to operate properly. This could take the form of system functions, CPCIs, or software functional requirements from Part I specifications (tailored to current user requirements). The next step is to decide if these software functions are already being evaluated in support of annex A objectives. If desired, separate subobjectives could be written for each major software function and cross-references made to the annex A objectives testing the subobjective. Alternatively, the software functions could be addressed in one subobjective and a cross-reference in the form of a matrix presented to indicate how the subobjective is being tested. For those software functions that are not already being

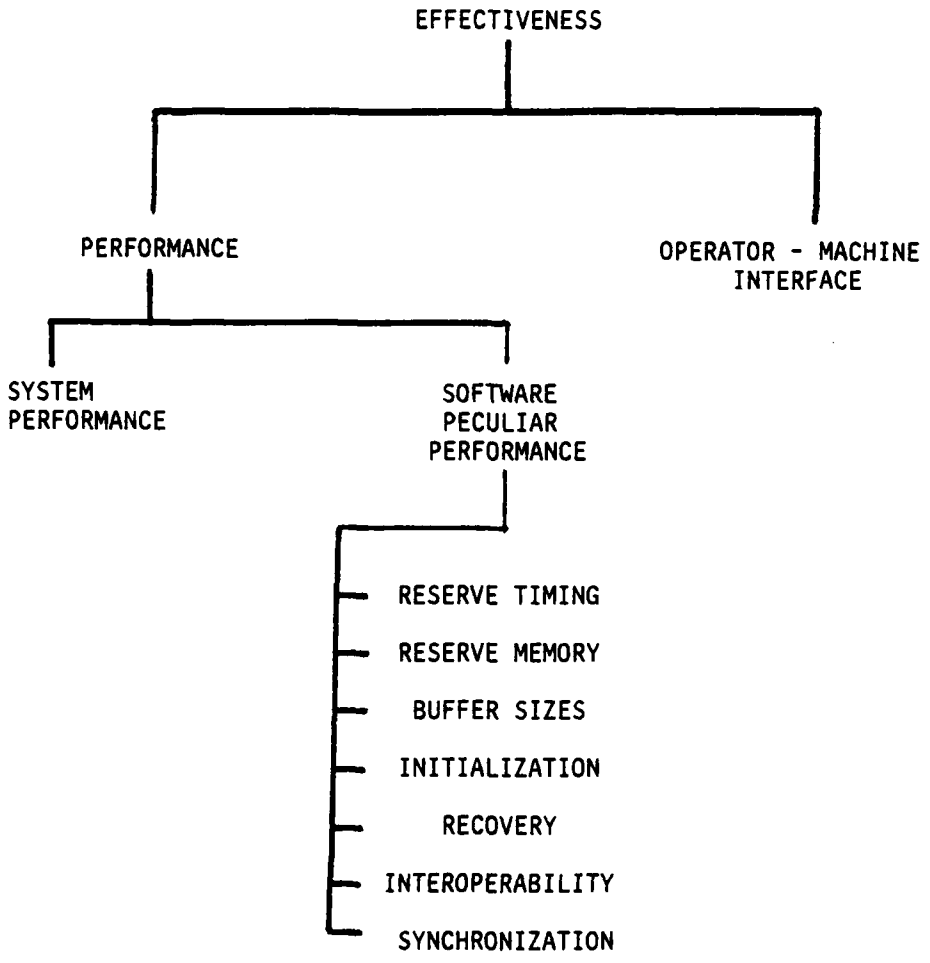

Figure 1. Software Operational Effectiveness Evaluation

evaluated by annex A, software peculiar subobjectives are written. For those subobjectives being tested by annex A objectives, the testing in support of those objectives is monitored by software personnel. All potential problems are investigated to ascertain the impact of software problems.

b) Method 2. The functions of the software are not defined, but a priori knowledge of the system indicates potential problems that bear close scrutinization. This approach demands the software test managers attendance at PDR, CRRs, etc., and a close liaison with other software personnel working the system. An established test team with competent software analysts is a big help. Software subobjectives are then written around potential problem areas. In addition, a general performance subobjective is written to assess the system impact of software problems.

c) Method 3. The functions of the software are not defined, nor is any attempt made to write individual software performance objectives. The software performance objective takes the form of software analysts tracking all operational tests and investigating all potential problems to ascertain the impact of software problems.

A2.3 Rating Software Problems Severity.

Regardless of which method is selected to arrive at software performance subobjectives, those subobjectives which address the analyses of software problems should require that software problems be rated in terms of their severity as follows:

a) Severity 1. An error which prevents the accomplishment of an operational or mission essential function, which interferes with an operator to the extent that the operator prevents the accomplishment

of an operational or mission essential function, or which jeopardizes personnel safety.

b) Severity 2. An error which adversely affects the accomplishment of an operational or mission essential function so as to degrade performance and for which no alternative workaround solution exists; or which interferes with an operator to the extent the operator adversely affects the accomplishment of an operational or mission essential function so as to degrade performance for which no alternative workaround solution exists. (Note: Reloading or restarting the program is not an acceptable workaround solution.)

c) Severity 3. An error which adversely affects the accomplishment of an operational or mission essential function so as to degrade performance and for which there is a reasonable, preferably predetermined alternative workaround solution, or which interferes with an operator to the extent that the operator adversely affects the accomplishment of an operational or mission essential function so as to degrade performance for which there is a reasonable workaround solution.

d) Severity 4. An error which is an operator inconvenience or annoyance and does not affect a required operational or mission essential function.

e) Severity 5. All other errors.

ATTACHMENT 3

APPENDIX 4

SOFTWARE OBJECTIVES TO SUPPORT OPERATIONAL SUITABILITY EVALUATIONS

A3.1 Suitability Subobjectives.

The operational suitability evaluation for software will typically be addressed under the categories of maintainability and usability as shown in figure 1. Figure 2 illustrates sample subobjectives. These subobjectives are discussed in the following paragraphs.

A3.2 Maintainability Subobjective.

Software maintainability subobjective addresses those characteristics of the software and the support facility which affect the ability of software maintenance personnel to modify the mission software to: a) correct errors, b) add system capabilities, c) delete features, and d) maintain compatibility with hardware changes. The two areas typically assessed in this evaluation are the maintainability of the mission software and the adequacy of the support facility.

The maintainability of the mission software is evaluated through the use of structured questionnaires covering the documentation and the source code.

The support facility evaluation combines a performance evaluation of the equipment and support software, an operator-machine interface assessment, and a maintainability evaluation for the support software.

A3.3 Usability.

Usability is the extent to which software designated to perform a support function is effective in performing that function and is

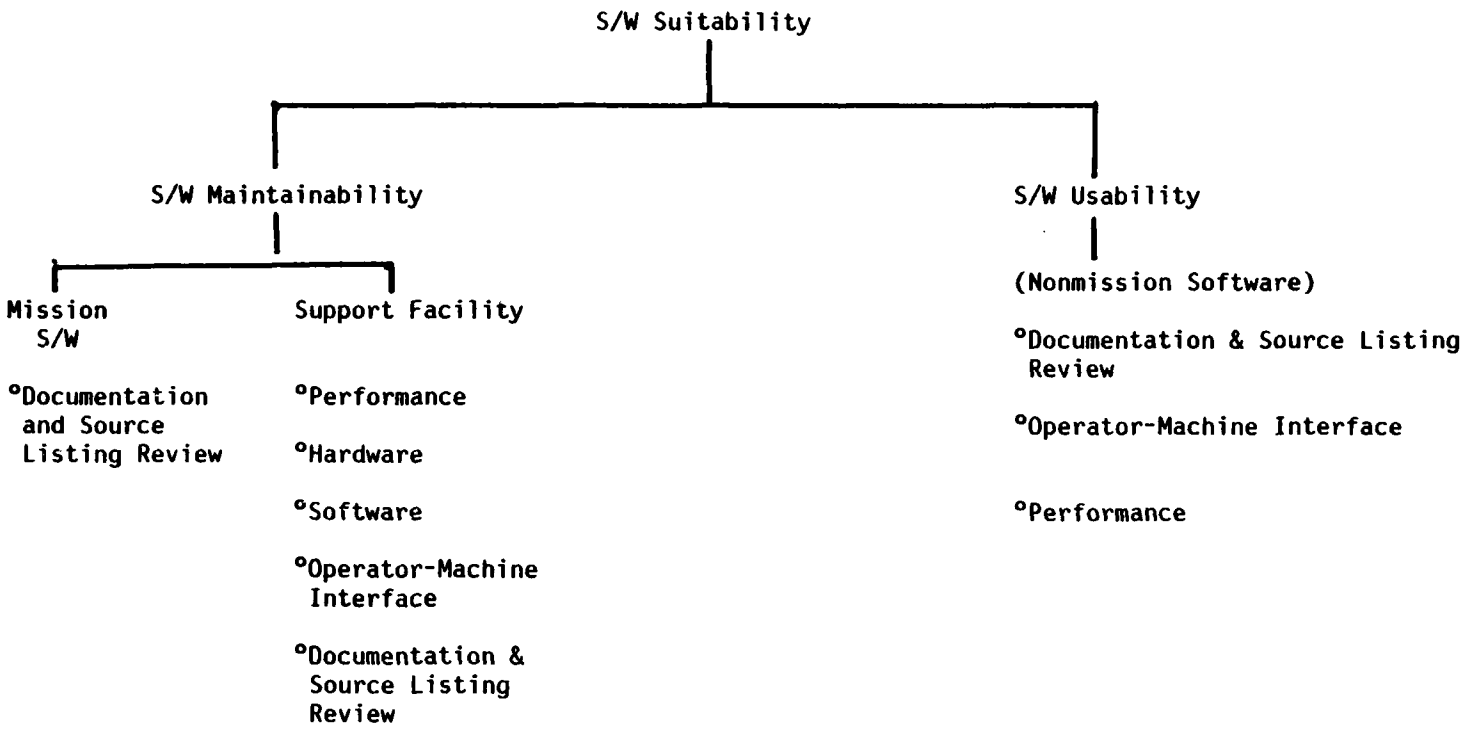

Figure 1. Software Suitability Evaluation

Objective 11. Evaluate the Operational Suitability of the CNCE Software Systems.

(A) Subobjective 11-1. Evaluate the Operational Software for Maintainability

(B) Subobjective 11-2. Evaluate Available Software Support Resources

} Maintainability

(C) Subobjective 11-3. Evaluate the Adequacy of Off-Line Diagnostics to Detect and Isolate Malfunctions in a Timely Manner

} Usability

Figure 2. Software Suitability Objectives Example

usable by the Air Force operator. This evaluation is typically an analysis of the adequacy and effectiveness of nonmission software (e.g., off-line diagnostics, ATE software) in terms of functional performance, operator-machine interface, and software maintainability.

APPENDIX 5

SOFTWARE ANNEX TEXT FOR STANDARD QUESTIONNAIRE USE

This appendix contains suggested software annex sections regarding the use of AFTEC standard questionnaires (software/ operator interface, maintenance). More information and procedural details are documented in appendices 6 and 7 of this handbook.

1.1 Subobjective SO-1.

Evaluate the software aspects of the operator-machine interface.

1.1.1 Measures of Effectiveness (MOE)/Evaluation Criteria.

MOE SO-1 is the average score of evaluator responses to the standard AFTEC Software Operator-Machine Interface Questionnaires.

The evaluation criteria are:

a) Threshold 3.30.

b) Standard 4.15.

c) Goal 5.00.

1.1.2 Methodology.

The software evaluators will complete standard, closed-form questionnaires. The evaluators will be provided a Software Operator-Machine Interface Evaluator's Handbook and a prebriefing on the evaluation procedures. Following the evaluation, a debriefing will be conducted to resolve uncertainties and to ensure that all evaluators have a common understanding of the questions. Although a standardized response set is required, the evaluators can include appropriate written comments.

1.1.3 Data Management.

1.1.3.1 Data Requirements.

Data required to complete the questionnaire are:

a) AFTEC's Software OT&E Guidelines, Volume IV, "Software Operator-Machine Interface Evaluator's Handbook."

b) Questionnaire Answer Sheet.

c) Operators manuals, etc., for subject equipment as determined applicable by the deputy for software evaluation.

However, no documents which are not deliverables to the government or are already permanently in the hands of the government will be considered during the evaluation.

1.1.3.2 Data Collection and Processing.

Completed answer sheets and comments will be collected by the deputy for software evaluation (DSE) and set to HQ AFTEC/TEBC, Kirtland AFB, New Mexico 87117. The answer sheets will be digitized and input to the Questionnaire Analysis Program (QAP) for data reduction and automated analysis.

1.1.3.3 Data Analysis.

Several data analysis functions can be accomplished by the analysis program at the request of the DSE. Some of the analysis features provided include:

a) Operator-machine interface computations.

1) Overall unweighted average score for each function.

2) Overall unweighted average average score by evaluator.

3) Unweighted average score for each factor.

4) Weighted average score by evaluator for each function.

5) Overall average weighted score.

6) Clear indication of products, test factors, and questions scoring below threshold.

b) Evaluation assessments.

1) Measure of evaluator agreement on each question.

2) User access to data base for specialized analysis.

The software test manager will perform a preliminary analysis of the automated reports incorporating comments provided by the evaluators. The automated reports and the software test manager's preliminary analysis will be returned to the deputy for software evaluation for further analysis and evaluation.

1.1.4 Evaluation.

The evaluation will be performed under the autnority of the test director, by the DSE with the assistance of the software evaluators and with the cognizance of the AFTEC software test manager. The questionnaire scores will be compared to the evaluation criteria. Additional investigation will be conducted in areas indicated by the questionnaire data as deemed necessary. Service reports will be prepared when necessary in accordance with established procedures.

1.2 Subobjective SO-2.

Evaluate the operational software for maintainability.

1.2.1 Measures of Effectiveness (MOE)/Evaluation Criteria.

MOE SO-2 is the average score of evaluator responses to the standard AFTEC software maintainability questionnaires.

The evaluation criteria are:

a) Threshold 3.30.

b) Standard 4.15.

c) Goal 5.00.

1.2.2 Methodology.

The software evaluators will complete standardized, closed-form questionnaires for each computer program being evaluated. Two questionnaires will be used: the Software Documentation Questionnaire and the Module Source Listing Questionnaire.

The Software Documentation Questionnaire set provides a measure of the extent to which the software design, reflected in the documentation, possesses good maintainability characteristics. The Module Source Listing Questionnaire set provides a measure of the extent to which the module source listings reflect a software implementation with good maintainability considerations. The evaluators will be provided a Software Maintainability Evaluator's Handbook and a prebriefing on the evalution procedures. A trial run will be conducted wherein each evaluator completes one Software Documentation Questionnaire and one Module Source Listing Questionnaire. Following the trial run, a debriefing will be conducted to resolve uncertainties and to ensure that all evaluators have a common understanding of the questions. The remainder of the questionnaires will be completed after the trial-run debriefing. Although the questionnaires use a standarized, closed-form response set to each question, an opportunity is provided for the evaluators to include written comments or expanded narratives as deemed appropriate.

Additional guidance and detailed procedures will be provided to the test team as a separate appendix.

1.2.3 Data Management.

1.2.3.1 Data Requirements.

Data required to complete the questionnaires are software documentation, software source listings, Software Maintainability Evaluator's Handbook, and answer sheets. Software documentation to be evaluated will normally include such items as computer program development specifications, computer software maintenance manuals, computer software test plans, version description documents, etc.

These documents may be in preliminary form at the time of the evaluation. However, no documents which are not deliverables to the government will be considered during the evaluation.

1.2.3.2 Data Collection and Processing.

Completed answer sheets and comments will be collected by the deputy for software evalution and sent to AFTEC/TEBC Kirtland AFB, New Mexico 87117. The answer sheets will be processed by an optical scanner and input to the Questionnaire Analysis Program (QAP) for data reduction and automated analysis.

1.2.3.3 Data Analysis.

Several data analysis functions can be performed by the analysis program at the request of the DSE. Some of the analysis features provided include:

a) Maintainability computations.
   1) Average score for each test factor and subfactor.
   2) Weighted score for documentation.
   3) Weighted score for each module.
   4) Weighted score across all modules.
   5) Weighted score of documentation and modules combined.
   6) Clear indication of products, test factors, and questions scoring below threshold.

b) Evaluation assessments.
   1) Measure of evaluator agreement on each question.
   2) Measure of reliability on each module question.
   3) User access to data base for specialized analysis.

c) Evaluation modification.
   1) Certain analysis parameters (e.g., agreement factor threshold) can be input.

2) Capability exists to delete evaluators, modules, or questions from the computation calculations, if necessary.

d) Cumulative data base.

1) A data base of all programs which have been evaluated can be maintained, updated, etc.

2) A cumulative report summarizing the evaluation results for each program in the data base can be output.

The software test manager will perform a preliminary analysis of reports from the analysis program incorporating comments/narratives provided by the evaluators. The program products and the software test manager's preliminary analysis will be returned to the deputy for software evaluation for further analysis and evaluation.

1.2.4.4 Evaluation.

The final evaluation is the responsibility of the test director, and the DSE, assisted by the software evaluators, and with the cognizance of the HQ AFTEC software test manager. The scores from the questionnaires will be compared to the evaluation criteria. The DSE will state whether the software is deficient, satisfactory, or excellent. Additional investigation will be conducted in areas indicated by the questionnaire data as advisable. Service reports will be prepared when necessary in accordance with established procedures.

APPENDIX 6

[Often Used As Appendix 1 to the
Software Annex of the Test Plan]

SOFTWARE MAINTAINABILITY DESIGN EVALUATION GUIDELINES

1.1 INTRODUCTION.

Software suitability evaluations consider two software qualities--usability and maintainability. The software products evaluated for maintainability are divided into three categories--source listings, documentation, and computer support resources. Each of these three categories is evaluated by considering certain maintainability attributes called test factors. This structure is shown in figure D-1-1. This appendix provides a standardized approach to evaluating the source listings and documentation for maintainability.

The methodology and procedures presented in this appendix provides a systematic approach to quantifying subjective data regarding some maintainability characteristics of software. The methodology capitalizes on the fact that software maintainability characteristics are essentially unchanged from program to program; therefore, a standardized evaluation technique can be used. Closed-form questionnaires, with opportunity for narrative comments, are used to guide the evaluator's thought processes to specific considerations and ensures that the same criteria are used by each evaluator. In addition, the terminology is standardized across a broad base of software maintainability evaluations.

The AFTEC software test manager and deputy for software evaluation have pre-established the relative weights of the source listing and documentation test factors.

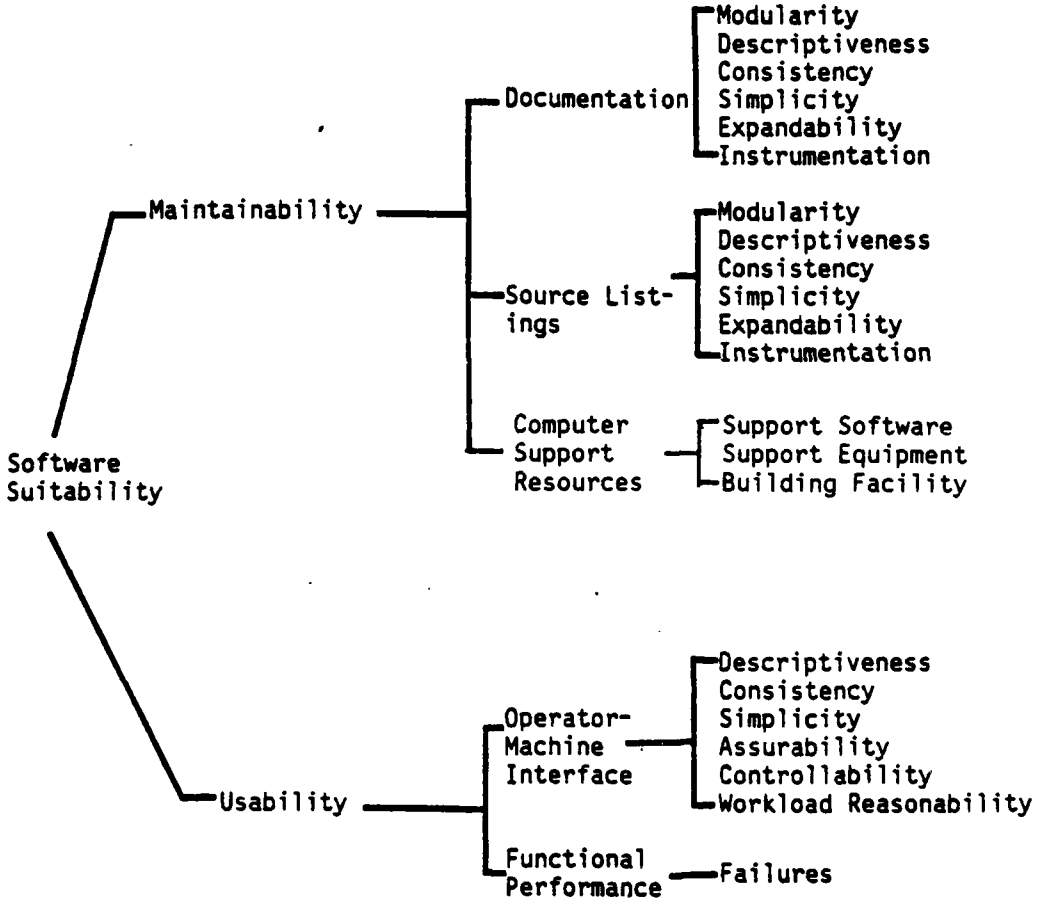

Figure D-1-1. Elements of Software Suitability Evaluation

This appendix includes definitions of applicable terms, measures of effectiveness/evaluation criteria, methodology and procedures, data management, and evaluation responsibilities. The standardized questionnaires and the list of programs and modules to be evaluated are attached to the appendix.

1.1.1 Assumptions.

The basic assumptions of this method are: 1) the source listing and documentation categories of software maintainability can be evaluated effectively by using the same criteria for all software; 2) the evaluators must be knowledgeable in software procedures, techniques, and maintenance but need not have detailed knowledge of the functional area for which a computer program is prepared; and 3) a random selection of modules within a computer program will be representative of the entire program.

1.1.2 Limitations.

The validity of this methodology may be limited if the evaluators do not have access to all of the software products they require.

1.1.3 Definitions.

Extended lists of definitions of software terminology are in the AFTEC Software OT&E Guidelines, Volume III, April 1980.

1.1.4 Environment.

The environment for this part of the evaluation will be a desk-top analysis of software products.

1.2 MEASURES OF EFFECTIVENESS (MOEs)/EVALUATION CRITERIA.

1.2.1 MOE.

MOEs are established at each of three levels: software quality (maintainability), selected software products (source listing, documentation), and test factors (modularity, descriptiveness, consistency, simplicity, expandability, instrumentation). At each level, the MOE is the weighted average of the scores associated with all questions applicable to that level. The Software Documentation Questionnaire (attachment 1) and the Module Source Listing Questionnaire (attachment 2) have one standard response set for all questions with the corresponding numerical values ranging from 1 (poorest score) to 6 (best score).

At the lowest level, the MOE for each test factor of each software product is the straight average of scores associated with all questions applicable to the given test factor. Therefore, all questions are weighted equally within a test factor.

At the next higher level, the MOE for each software product (documentation and source listing) is the sum of the products of the test factor relative weight and test factor raw score as determined above. The relative weights of the six test factors for the documentation evaluation sum to one, as do the weights of the six test factors for the source listing evaluation.

At the highest level, the MOE of the software quality being evaluated (maintainability) is the sum of the products of each software product score and the associated relative weight. The computer support resources are not addressed in this appendix, but are included in the maintainability MOE. Therefore, the relative weights for documentation and source listings do not total to one.

The relative weights for each of the test factors and for documentation and source listing are presented in tables D-1-1, D-1-2, and D-1-3, for guidelines to the test team.

1.2.2 Evaluation Criteria.

The deputy for software evaluation (DSE) and the HQ AFTEC software test manager have determined the evaluation criteria

Table D-1-1

Source Listing Questionnaire Test Factor Weights

Source Listing Test Factor	Factor Weight
Modularity	.15
Descriptiveness	.22
Consistency	.18
Simplicity	.20
Expandability	.12
Instrumentation	.13
Category Score (Total)	1.00

Table D-1-2

Documentation Questionnaire Test Factor Weights

Documentation Test Factor	Factor Weight
Modularity	.14
Descriptiveness	.25
Consistency	.18
Simplicity	.18
Expandability	.12
Instrumentation	.13
Category Score (Total)	1.00

Table D-1-3

Maintainability Test Factor Weights

Maintainability Test Factor	Factor Weight
Documentation	.35
Source Listing	.40
Computer Support Resources (determined elsewhere)	.25
Maintainability (Total)	1.00

(threshold, standard, and goal). These evaluation criteria are based on numerical values assigned to each response of a standardized questionnaire response set as follows:

A. Completely Agree (6 points).
B. Strongly Agree (5 points).
C. Generally Agree (4 points).
D. Generally Disagree (3 points).
E. Strongly Disagree (2 points).
F. Completely Disagree (1 point).

The evaluation criteria are:

Goal	5.00
Standard	4.15
Threshold	3.30

HQ AFTEC/TEBC has established guideline relative weights and evaluation criteria for a "typical" software package. HQ AFTEC software test managers, in conjunction with the DSE, are authorized to deviate from these guidelines values when the support concept or specific functional application so dictates. The guideline values were shown in tables D-1-1, D-1-2, and D-1-3 above.

1.3 METHODOLOGY/PROCEDURES.

The test and evaluation methodology consists of completing standardized, closed-form questionnaires by five or more software evaluators for each computer program being evaluated. Two questionnaires are used: the Software Documentation Questionnaire (attachment 1) and the Module Source Listing Questionnaire (attachment 2).

The Software Documentation Questionnaire is completed once by each evaluator for complete each computer program being evaluated. The completed questionnaire set provides a measure of the extent to which the software design, reflected in the documentation, possesses good maintainability characteristics. In addition, information is gathered on the format and organization of the software documentation. The Software Documentation Questionnaire consists of a total of 83 questions addressing various aspects of modularity, descriptiveness, consistency, simplicity, expandability, and instrumentation.

The Module Source Listing Questionnaire is completed once by each evaluator for each of approximately 10 to 25 percent of the modules of the computer program. The modules to be considered in the evaluation are selected at random and are assumed to be representative of the complete set of computer program modules (see procedures below). The completed questionnaire set provides a measure of the extent to which the module source listings reflect a software implementation with good maintainability considerations. In addition, the Module Source Lising Questionnaire contains questions for the evaluation of the consistency between software documentation and the source listings. The questionnaire consists of 89 questions concerning software modularity, descriptiveness, consistency, simplicity, expandability, and instrumentation.

The test methodology requires a minimum of five evaluators who are knowledgeable in software procedures, techniques, and maintenance. Five evaluators are necessary to provide statistical confidence that the test data provides a valid measure of software maintainability. The evaluators will be provided a Software Maintainability Evaluator's Handbook and a prebriefing on the evaluation procedures. A trial run will be conducted wherein each evaluator completes one Software Documentation Questionnaire and one Module Source Listing Questionnaire. Following the trial run, a debriefing will be conducted to resolve uncertainties and to ensure that all evaluators have a common understanding of the questions. The remainder of the questionnaires are completed after the trial run debriefing.

Although the questionnaires use a standardized, close-form response set to each question, an opportunity is provided for the evaluators to include written comments/expanded narratives as deemed appropriate.

The deputy for software evaluation will determine the evaluators for each computer program which will be evaluated. A minimum of five evaluators will complete questionnaires on each computer program being evaluated. It is desirable, but not mandatory, that one evaluation team evaluates all the software for a given OT&E. Each evaluator must be knowledgeable in software procedures, techniques,

and maintenance, but need not have detailed knowledge of the functional application of the software. It is preferable that evaluators are persons who will be responsible for maintaining some part of the software for the system undergoing OT&E.

In general, two levels of software structure are considered in each evaluation. These two levels are program level (the higher) and module level (the lower). A given OT&E may separately address a number of computer programs at the higher level. One Software Documentation Questionnaire will be completed for each program by each evaluator. At the module level, a specific number of Module Source Listing Questionnaires will be completed by each evaluator. A tentative list of programs and modules to be evaluated is shown in table D-1-4. This table will be updated before the start of the evaluation to reflect the most current software structure and the specific programs and modules selected for evaluation. The following minimum requirements will be reflected in the table:

a) All programs which will be routinely maintained in-house by an Air Force agency will be evaluated.

b) The number of modules to be evaluated within each program will be at least 10 percent but not more than 35 modules.

c) If the program has an executive module, it will be selected for evaluation.

Each evaluator will complete one (and only one) Software Documentation Questionnaire and one (and only one) Module Source Listing Questionnaire as a trial run. A trial run debriefing will be conducted by HQ AFTEC personnel. During this briefing, evaluators will have questions answered and uncertainties resolved. The debriefing also provides assurance that all evaluators have a common understanding of the questions. After the trial run debriefing, evaluators will have the opportunity to change answers as necessary. The remainder of the questionnaires will be completed after the trial run debriefing. The specific computer program and modules within that program to be evaluated for the trial run will be identified in table D-1-4.

Table D-1-4

Computer Programs, Modules to be Evaluated

(TBD)

The responses (answers) for all questions will be entered on a General Purpose (National Computer Services) Answer Sheet. In addition to a mandatory response from the standard response set, each evaluator has the opportunity to provide written comments or narrative expansions on each question. All comments/narratives will be provided to the software test manager. Detailed instructions for completing the General Purpose (NCS) Answer Sheets are included in the Software Maintainability Evaluator's Handbook which will be provided to each evaluator.

1.4 DATA MANAGEMENT.

1.4.1 Data Requirements.

Data required to complete the questionnaires are software documentation, software source listings, Software Documentation Questionnaires, Module Source Listing Questionnaires, Software Maintainability Evaluator's Handbook, and General Purpose (National Computer Services) Answer Sheets. Software documentation to be evaluated will normally include such items as computer program development specifications, computer software maintenance manuals, computer software test plans, version description documents, etc. These documents may be in preliminary form at the time of the evaluation. However, no documents will be considered during the evaluation which are not deliverables to the government.

1.4.2 Data Collection/Processing.

Completed answer sheets and comments will be collected by the deputy for software evaluation and sent to AFTEC/TEBC, Attn (software test manager's name) Kirtland AFB, New Mexico 87117. The answer sheets will be processed by an optical scanner and input to the Questionnaire Analysis Program (QAP) for data reduction and automated analysis.

1.4.3 Data Analysis.

Several data analysis functions can be accomplished by the QAP at the request of the DSE. Some of the analysis features provided include:

a) Maintainability computations.
   1) Average score for each test factor and sub-factor.
   2) Weighted score for documentation.
   3) Weighted score for each module.
   4) Weighted score across all modules.
   5) Weighted score of documentation and modules combined.
   6) Clear indication of products, test factors, and questions scoring below threshold.

b) Evaluation assessments.
   1) Measure of evaluator agreement on each question.
   2) Measure of reliability on each module question.
   3) User access to data base for specialized analysis.

c) Evaluation modification.
   1) Certain analysis parameters (e.g., agreement factor threshold) can be input.
   2) Capability exists to delete evaluators, modules, or questions from the computational calculations, if necessary.

d) Cumulative data base.
   1) A data base of all programs which have been evaluated can be maintained, updated, etc.
   2) A cumulative report summarizing the evaluation results for each program in the data base can be output.

The software test manager will perform a preliminary analysis of reports from the QAP incorporating comments/narratives provided by the evaluators. The QAP products and the software test manager's preliminary analysis will be returned to the deputy for software evaluation for further analysis and evaluation.

1.5 EVALUATION.

The DSE will review the results of the analysis to identify whether the software maintainability aspects of the system are unsatisfactory, satisfactory, good, or excellent. This assessment will be determined based on whether the maintainability evaluation results meet the established threshold, standard, or goal criteria. Where deficiencies or needed improvements are identified, they will be investigated for possible discussion in the final report.

APPENDIX 7

[Often Used as Appendix 2 to the
Software Annex of the Test Plan]

SOFTWARE OPERATOR-MACHINE INTERFACE DESIGN EVALUATION GUIDELINES

1.1 INTRODUCTION.

The design of software to accommodate interactions between the system operators and the machine is an important consideration for any embedded computer system. Software operator-machine interface evaluations will be undertaken in both the effectiveness and the suitability areas. This relationship is shown in figure D-1-1. The operator-machine interface will be evaluated by considering certain attributes called test factors. This structure is shown in figure D-1-2. The methodology and procedures presented in this appendix provide a systematic approach to quantifying subjective data regarding operator-machine interface characteristics of software. The methodology capitalizes on the fact that software interface characteristics are essentially unchanged from program to program; therefore, a standardized evaluation technique can be used. Closed-form questionnaires, with opportunity for comments, will be used to guide the evaluator's thought processes to specific considerations and to ensure that the same criteria are used by each evaluator.

This appendix addresses MOEs, evaluation criteria, methodology, procedures, data management, and evaluation responsibilities.

1.1.1 Assumptions.

The basic assumptions of this method are:

a) Those characteristics which contribute positively to operator-machine interfaces are similar for all embedded computer systems.

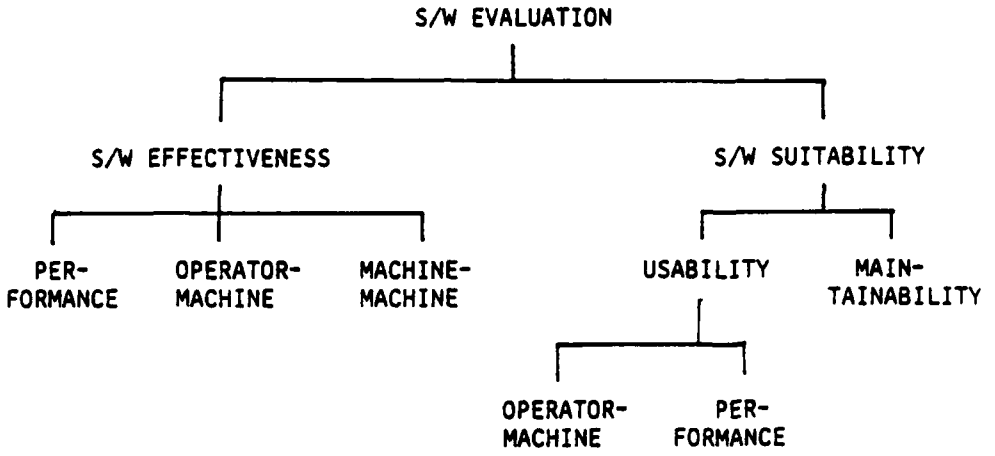

Figure D-1-1. Software Operator-Machine Interface Questionnaire (SOMIQ) Environment

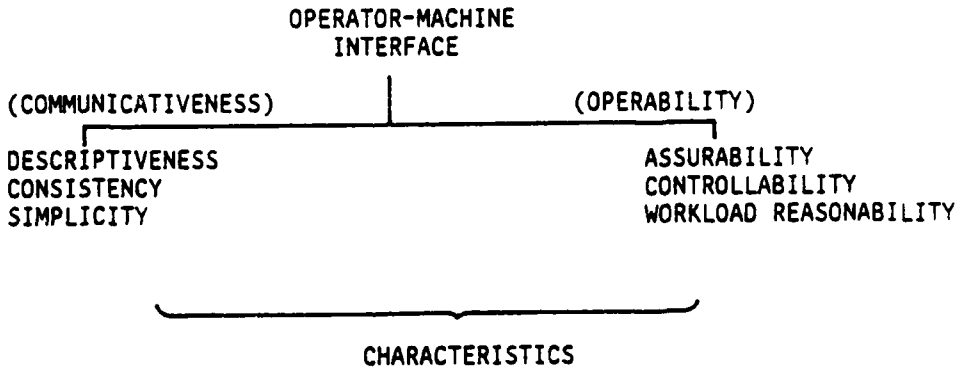

Figure D-1-2. Software Operator-Machine Interface Questionnaire Conceptual Model

b) The evaluators must be knowledgeable in system procedures but need not have detailed knowledge of the computer program design.

c) A random selection of functions performed by a computer program will be representative of the entire program.

1.1.2 Limitations.

The embedded computer system and associated peripherals must function reasonably correctly. Otherwise the operator will use so much time with abnormal procedures that any normal positive attributes found under operator-machine interface will be negated.

1.1.3 Definitions.

Extended lists of definitions of software terminology are in the AFTEC Software OT&E Guidelines, volume IV, July 1980.

1.1.4 Environment.

Evaluators will probably not be able to complete the questionnaire while actually operating the system. The evaluator completing the questionnaire should observe another operator over-the-shoulder while formulating answers.

1.2 MEASURES OF EFFECTIVENESS (MOEs)/EVALUATION CRITERIA.

1.2.1 MOEs.

MOEs are established at each of three levels: 1) overall system (top level), 2) selected software functions, (e.g., mission data preparation, calibration), and 3) test factors (e.g., assurability, descriptiveness).

At each level the MOE is the average score from all questions applicable to that level.

At the lowest level the MOE for each test factor of each software function is the unweighted average score from all questions within the given test factor; therefore, all questions are weighted equally within a test factor.

At the next higher level the MOE for each software function is the weighted average of the test factor raw scores as determined above. The relative weights of the six test factors for the Software Operator-Machine Interface Questionnaire (SOMIQ) evaluation sum to one.

At the highest level the MOE of the operator-machine interface is the weighted average of the scores for all software functions under evaluation.

The relative weights for each of the test factors and each of the software functions is presented in table D-1-A1 as guidelines for the test team.

Table D-1-A1

Test Factor Weights

Test Factor	Factor Weight
Assurability (15/71)	.21
Controllability (13/71)	.18
Workload Reasonability (14/71)	.20
Descriptiveness (12/71)	.17
Consistency (7/71)	.10
Simplicity (10/71)	.14
Total (71/71)	1.00

1.2.2 Evaluation Criteria.

The evaluation criteria are based on numerical values assigned to each response of a standardized questionnaire response set as follows:

A. Completely Agree (absolutely no doubt) (6 points)
B. Strongly Agree (5 points)
C. Generally Agree (4 points)

D. Generally Disagree (3 points)

E. Strongly Disagree (2 points)

F. Completely Disagree (absolutely no doubt) (1 point)

The evaluation criteria are:

a) Goal 5.00

b) Standard 4.15

c) Threshold 3.30

1.3 METHODOLOGY/PROCEDURES.

The test and evaluation methodology consists of completion of the questionnaires by five software evaluators for each computer function being evaluated.

Five evaluators are recommended to provide statistical confidence that the test data provides a valid measure of software operator-machine interface characteristics. The evaluators will be provided an evaluator's guideline handbook and a prebriefing on the evaluation procedures. A trial run will be conducted wherein each evaluator completes one SOMIQ. Following the trial run a debriefing will be conducted to resolve uncertainties and to ensure that all evaluators have a common understanding of the questions. The remainder of the questionnaires will be completed after the trial run debriefing. Although a standardized response set is required, the evaluators can include appropriate written comments.

The deputy for software evaluation will assign the evaluators for each computer function to be evaluated. The recommended five evaluators will complete questionnaires on each computer function evaluated. One evaluation team should evaluate all the software for similar computer functions. Evaluators should be persons who will be responsible for maintaining or operating some part of the software for the system undergoing OT&E.

1.4 DATA MANAGEMENT.

1.4.1 Data Requirements.

Data required to complete the questionnaire are:

a) AFTEC's Software OT&E Guidelines, Volume IV, "Software Operator-Machine Interface Evaluator's Handbook."
b) Questionnaire answer sheet.
c) Operators manuals, etc., for subject equipment as determined applicable by the test team.

1.4.2 Data Collection/Processing.

Completed questionnaire answer sheets and any written comments will be collected by the test team and sent to AFTEC/ TEBC, Attn (software test manager's name), Kirtland AFB, New Mexico 87117. Answers will be input to the QAP for data reduction and automated analysis.

The AFTEC software test manager is responsible for a preliminary analysis of the automated reports incorporating comments provided by the evaluators. The automated reports and the software test manager's preliminary analysis will be returned to the test team for final analysis and evaluation.

1.4.3 Data Analysis.

Several data analysis functions can be accomplished by the SOMIQ analysis program at the request of the DSE. Some of the analysis features provided include:

a) Operator-machine interface computations.
    1) Overall unweighted average score for each function.
    2) Overall unweighted average score by evaluator.
    3) Unweighted average score for each factor.
    4) Weighted average score by evaluator for each function.
    5) Overall average weighted score.
    6) Clear indication of products, test factors, and questions scoring below threshold.

b) Evaluation assessments.

1) Measure of evaluator agreement on each question.

2) User access to data base for specialized analysis.

1.5 EVALUATION.

The DSE will review the analysis results to identify the degree to which the software operator-machine interface meets the established threshold, standard, and goal criteria. Where deficiencies or needed improvements are identified, they will be investigated for possible discussion in the final report.

APPENDIX 8

SAMPLE DOCUMENT REVIEW

Reply to: TEB

Subject: ABCD TEMP Coordination (Your Ltr, 2 Jan 81)

To: HQ AFTEC/TEX

Your letter dated 2 Jan 81 requested coordination only on the ABCD TEMP, but in reviewing the document for compliance with our past comments we feel there are several areas concerning software identification and testing that are unsatisfactory. Coordination on the ABCD TEMP by this office is contingent on satisfactory resolution of the following software concerns. We must emphasize that resolution of the following concerns in the ABCD TEMP may alleviate a number of potential software test and evaluation issues (particularly with software integration and management responsibilities) and will force program office personnel to address software issues they have been ignoring.

a. Page I-3, paras 1b(1)(c) and 1b(2)(c). These two subparagraphs are typical of the testing descriptions provided in this TEMP. They refer only to the testing of hardware with no indication of software testing. Recommend level of software testing be added to all paragraphs that relate to subsystem or systems level testing involving software.

Rationale: Nowhere in this document is any reference made to actual phases or objectives for software testing (except for a very brief and inadequate reference to IV&V in paragraph 2a(1), page III-3). Both DOD directives 5000.29 and 5000.3 emphasize the need to identify software testing and validation requirements to reduce system integration risks.

b. Page I-5, para 3a(1) and page 1-15, para 6. Adaptive reprogramming is identified as one of the few software related operational requirements, but there is no definition of this activity nor any identification of adaptive reprogramming as a technical characteristic (para 6, page I-15) for the ABCD system. Recommend identification of the operational criteria for which adaptive reprogramming will be evaluated, including a definition of the activity.

Rationale: It is not possible to generate an adequate test objective for adaptive reprogramming evaluation without knowing its operational need and test criteria.

c. Page I-6, para 4. One of our major concerns is the total lack of ABCD software descriptions and test objectives. To resolve the first concern, we recommend an additional paragraph, 4e, be added with two subparagraphs to address ABCD software. As a minimum, a description of the ABCD software should include any contractual design requirements or known deviations pertaining to software and the type of control structure (executive module versus interrupt processing). Of particular importance is an identification of the anticipated number of software baselines related to the various ABCD configurations. Paragraph 4 emphasizes hardware and technology commonality without identifying related software.

Rationale: The adequacy of software design and integration is a major component of the ABCD systems and must be understood by all agencies concerned with test and evaluation. A detailed discussion of ABCD software is not required in the TEMP but enough detail should be provided to emphasize the impact of software on system performance and maintenance and the need for adequate software/integration testing.

d. Page I-9, para 5a. There is no reference in this section to evaluating effectiveness of ABCD software. Recommend a sentence be added to this paragraph indicating that the contribution of software to ABCD performance will also be evaluated during all phases of testing for the various levels of integration.

Rationale: Software is a major contributor to system performance and it often times singled out as the major contributor to poor systems performance. It is only fair that it be recognized for test planning as an important performance consideration and a major area of concern.

e. Page I-15, para 5b. Recommend an additional subparagraph 5b(13), be added to address software suitability evaluations. In particular, operational software for ABCD will be evaluated for maintainability and support software/support equipment software will be evaluated for performance adequacy in the support area, to include operator interface adequacy, and software maintainability.

Rationale: Suitability of software is a significant area of concern once the system becomes operational and should be addressed as an operational suitability test issue.

f. Page I-20, para 7b(3). Change the third sentence to read: "Have the ABCD maintainability, reliability, and software suitability requirements been met...?"

Rationale: Identification of software suitability as an ABCD suitability issue.

g. Page III-3, para 2a(2). This paragraph provides the only reference in the TEMP to the utilization of independent verification and validation to evaluate software, but this section only covers the period November 1982-November 1984 and doesn't give any indication of the scope of the IV&V to be performed. Recommend an additional paragraph be added to each phase of DT&E testing involving software that describes the level of IV&V and methods to be used (i.e., new paras 2a(5), 2c(5), 2e(5)). It is also conceivable that some IV&V will be required beyond November 1984 during DT-IIIB to assess the final software configuration.

Rationale: IV&V is a necessary technique for evaluating software design and development adequacy and progress, but its contribution is highly dependent on the level of effort contracted for, the tools utilized to perform IV&V, and the duration of the IV&V. The TEMP does not address IV&V in sufficient detail to determine its intent and support of T&E.

h. Page III-5, para 2c(3). The software verification and validation (V&V) referenced in this paragraph is assumed to be development contractor V&V not IV&V. Will the IV&V contractor utilize an environmental simulator?

i. Page IV-7, para 3. Add an additional paragraph, 3h, to read "The availability of current and complete software documentation, to include source listings."

_____, Colonel, USAF
Chief, Computer/Support Systems Div

MR: We may have been delinquent in providing these significant software T&E concerns on an earlier version of this TEMP but whether we let the opportunity slip or not does not detract from the significant deficiency in this TEMP in addressing ABCD software T&E.

APPENDIX 9

SAMPLE SOFTWARE PORTION OF FINAL REPORT

Objective 1.

Evaluate the operational suitability of the A-10 OFT software.

Subobjective 1-1.

Determine the A-10 OFT software maintainability.

Method.

The deputy for software evaluation (DSE) and the software evaluators evaluated eleven software modules selected as representative of the A-10 OFT software. In the selection of modules, consideration was given to those most likely to require maintenance during the life of the system. Both the real time and the nonreal time computer program configuration items were evaluated. Standard AFTEC questionnaires, designed to measure the presence of desirable maintainability characteristics in the documentation and source code, were used. In addition, questionnaires specially devised to evaluate the computer support resources needed to maintain aircrew training device software were completed.

The standard AFTEC questionnaires have been used to evaluate software maintainability characteristics in systems over the past three and one-half years. Two closely-related questionnaires were used on the A-10 OFT evaluation; one designed to provide data on the software documentation, and one which focuses on the characteristics of the source code. Based on past evaluations there was high confidence that the completed questionnaires would yield results indicative of actual field experience.

The questionnaires have a six point response scale where 6 is the highest possible score and 1 is the lowest. Questions were

grouped into several test factors, and the scores for all questions applicable to a given test factor were averaged to obtain the score for that factor. Each factor was assigned a relative weight, based on its importance, to arrive at an overall score. Thus, the measures of effectiveness were straight averages for test factors and the weighted averages of these test factor scores for documentation and source listing. The weights used on the A-10 OFT evaluation were the same as had been used on a number of other evaluations. The threshold (3.3), standard (4.15), and goal (5.0) (on the six point scale) were also the same as had been used for previous evaluations.

Computer support resources consist of support software, support equipment, and the support facility (building). The questionnaires for computer support resources were more subjective than the standard questionnaires. However, they had been applied to the same organically supported simulators mentioned earlier and the results compared with field experience. A six point scale was also used for evaluating computer support resources. A straight average of the scores for each test factor (support software, support equipment, and building) was calculated to obtain an overall score for computer support resources.

Results and Discussion.

The scores for software maintainability are show in table XX. This table provides the evaluation results for documentation, source listings, and computer support resources along with the scores for the test factor under each of these categories. These were combined into an overall maintainability score which is also provided. The threshold, standard, and goal are included for ease of comparison.

The maintainability characteristics of the documentation evaluation resulted in an overall rating below the threshold and thus deficient. This was due to the very low rating on instrumentation, modularity, and descriptiveness. The low rating on descriptiveness and modularity were primarily due to deficiencies in part 2 of volume 4 program descriptions titled "Trainer Program Operation Overview." Updates and additions to this section will be part of a

Table XX

Software Maintainability Assessment

Item Rated	Score	Evaluation Criteria Threshold	Standard	Goal
Maintainability	3.84*	3.30	4.15	5.00
Documentation	3.27**			
Modularity	2.64**			
Descriptiveness	3.25*			
Consistency	3.88			
Simplicity	4.48			
Expandability	3.64			
Instrumentation	1.93**			
Source Listings	4.15			
Modularity	5.04			
Descriptiveness	3.83*			
Consistency	4.06			
Simplicity	4.47			
Expandability	4.41			
Instrumentation	2.59**			
Computer Support Resources	3.60	2.80	3.40	4.50
Support Software	3.40*	3.30	3.70	4.70
Support Equipment	3.30	3.00	3.30	3.70
Building	4.20	2.00	3.30	5.00

*Between Threshold and Standard
**Below Threshold

later revision. A brief review of the latest revision indicates a substantial improvement in the overall documentation rating.

The maintainability characteristics of the source code were rated satisfactory. The weak areas were the instrumentation and descriptiveness factors. The instrumentation low score was due to a lack of in-program test indicators. The descriptiveness low score was due to insufficient information in preface blocks and comment fields.

The computer support resources were rated satisfactory. One test factor, support software, scored below the standard.

Conclusion.

The A-10 OFT software is maintainable.

Recommendations.

All future changes to the source code should incorporate commentary to include preface blocks and to explain the objectives and purpose of the section of code.

It is recommended that a follow-on test and evaluation be performed on the software documentation package when it is delivered 60 days after unit number 1 is ready for training. It is also recommended that the simulator update program (SUMP) be evaluated at the time of delivery.

Subobjective 1-2.

Evaluate the usability of the A-10 software packages.

Method.

This evaluation was designed to determine the usefulness and suitability of computer programs which support simulator operations. Four computer program functional applications were selected for evaluation: daily operations, mission data preparation, diagnostics, and calibration. Questionnaires unique to aircrew training devices

were devised specifically for each of these four applications. They were very similar to those used for the computer support resources evaluation discussed earlier. In addition, forms to be used for recording observer comments during the use of diagnostics and calibration programs addressed load and run problems, understandability, and data interpretation. All questionnaires were scored on a zero to five point scale much the same as was used for the maintainability evaluation. The measures of effectiveness were the average scores on this five point scale. Threshold, standard, and goal values were determined in advance by the DSE and are presented in table XXX. The questionnaires were completed by the software evaluators during the in-plant testing.

Table XXX

Usability Evaluation Criteria

Functional Application	Threshold	Standard	Goal
Daily Operation	3.0	3.9	5
Mission Data Preparation	3.0	4.0	5
Diagnostics	2.9	3.7	5
Calibration	2.7	3.7	5

Results.

The usability factors of daily operation and mission data preparation were undetermined. The technical orders (TOs) required to complete the evaluation of these factors were not available at the time of the test. The TOs will be available 120 days after "ready for training" date on unit number two.

The diagnostic and calibration test factor results were combined and were rated satisfactory.

A service report (SR) was not written on the TOs for daily operation and mission data preparation factors being unavailable because the simulator will be contractor operated and maintained for a period of two years after the ready for training date.

Conclusions.

The computer programs to support the diagnostic and calibration of the A-10 OFT are satisfactory with no additional testing required.

Follow-on test and evaluation on the daily operation and mission data preparation test factors is warranted when the TOs become available. After two years the A-10 OFT will be Air Force operated and maintained.

Recommendations.

Recommend that follow-on test and evaluation of the daily operation and mission data factors of usability be conducted after TOs are delivered and prior to the Air Force assuming total operation and maintenance responsibilities of the A-10 OFT.

Objective 2.

Evaluate the operational effectiveness of the A-10 OFT software.

Method.

No separate evaluation criteria (threshold, standard, or goal) was used for the evaluation of software aspects of fidelity, training capability, instructional features, and electronics warfare instructional features. Rather, the evaluation criteria for operational effectiveness include the software contribution.

During the in-plant QOT&E mission testing, the operation of the entire simulator system was closely monitored, and test and mission events were logged. This log together with all test descriptions (TDs) written during QOT&E were reviewed to determine if software design and implementation were the cause of poor fidelity, training capability, or instructional capability.

The instructor operator station (IOS) was evaluated using a standard AFTEC software questionnaire, the Software Operator Machine Interface Questionnaire (SOMIQ). The SOMIQ was used to

determine the utilization and effectiveness of the IOS by the instructor pilots (IPs) as related to training capability and task performance. It addresses the software impact of the IP actions with the IOS rather than human factors considerations. The SOMIQ was completed by the IPs during QOT&E mission testing.

The questionnaires have a standard six point response scale where 6 is the highest possible score and 1 is the lowest. Questions were grouped into several test factors, and the scores for all questions applicable to a given test factor were averaged to obtain the score for that test factor. Each factor was assigned a relative weight, based on its importance, to arrive at an overall score. Thus, the measures of effectiveness were straight averages for test factors, and these test factor scores were averaged to determine the SOMIQ score. The threshold (3.30), standard (4.15), and goal (5.0) were the same as had been used for previous evaluations.

Results.

(Insert software effectiveness write up after on-site testing.)

The software interface of the IOS and the IP was satisfactory with improvements required. The descriptive test factor was rated below the threshold. An instructors manual, or handbook, was not available for the IPs to use during the evaluation. There had been no formal training of the IPs on the IOS just prior to mission testing.

The scores from the SOMIQ are shown in table XXXX. This table provides the evaluation results of the eight IPs who performed the evaluation.

Conclusion.

(Insert software effectiveness writeup after on-site testing.)

The software effectiveness of the IOS is satisfactory for training capabilities with improvements required.

Recommendations.

(Insert software effectiveness write-up.)

Formal training should be given the IPs just prior to QOT&E mission testing. An IOS handbook/manual must be available for the IPs during test.

Table XXXX

Software Operators Machine Interface Assessment

Item Rated	Score	Evaluation Criteria Threshold	Standard	Goal
Operator-Machine Interface	3.86*	3.30	4.15	5
Factors				
Assurability	3.93*			
Controlability	4.36			
Workload	3.82*			
Descriptiveness	3.20**			
Consistency	3.87*			
Simplicity	3.91*			

*Between Threshold and Standard
**Below Threshold

It is also recommended that a follow-on evaluation of the IOS handbook/manual be conducted by the user after the A-10 OFT is ready for training.

APPENDIX 10

OPERATIONAL TEST AND EVALUATION OF SOFTWARE

by Walter G. Murch, Major, USAF

Air Force Test and Evaluation Center
Kirtland AFB, NM

ABSTRACT

The Air Force Test and Evaluation Center (AFTEC) has developed techniques for the Operational Test and Evaluation (OT&E) of Weapon System Software since 1976. The objectives of the evaluation of the software are to determine the operational effectiveness and operational suitability of the software to support mission operations. Operational effectiveness connotes the capability of the software to perform its intended functions in the operational environment while operational suitability connotes the degree to which the software supports the mission and is maintainable. This paper discusses the history of software OT&E and the software attributes which are important to the user and maintainer, and describes the AFTEC approach to software OT&E.

HISTORY AND POTENTIAL

Operational Test and Evaluation.

The Bolendar Committee in their September 1970 "Report on Operational Test and Evaluation"[6] defined five broad objectives of operational test and evaluation (OT&E).

a) Determine operational suitability or acceptability of new or improved weapons systems, subsystems, and equipments.
b) Determine the compatibility of new and improved weapon systems, subsystems, and equipment with the operational environment within which they must operate.
c) Determine the feasibility and suitability of new operational concepts, doctrine, tactics, techniques and procedures.
d) Determine the effectiveness of operational capabilities.
e) Obtain baseline data to support future operational requirements, reconfiguration of force structure, and realignment of roles and missions.

As such the OT&E provides a bridge between the development test and evaluation (DT&E) and operational usages. The effect of this is evident if the life cycle of a system development is considered. Figure 1[12] outlines this life cycle. Note especially that the user community defines a mission need which, when validated, is the baseline for development of the system. Between the time that this need is developed and the time that it is given over to the users and support agencies, substantial activity takes place and many agencies are involved. A partial listing of these activities includes:

a) Operations and support concept development.
b) Specification development.
c) Concept development contracts written.
d) Prototype contractor chosen.
e) Full scale development contract enacted.
f) Detailed specification development.
g) Requirement scrubbing (to match resources).
h) Requirement scrubbing (to match technology).
i) Operations and support concept revision (to accommodate requirement change).

Some of the agencies involved include: Department of Defense, Department of Air Force, Air Force System Command, system program office, Air Force Logistics Command, General Accounting Office (GAO), congress, Office of Management and Budget - and perhaps the user and supporter. All of these agencies massage the many contractor innovations. The final product should reflect the needs of the operational and support agencies.

The DT&E of the end item is intended to verify compliance with the specifications that evolved during the development cycle. As such, DT&E includes engineering tests specifically designed to evaluate the system against these specifications. The role of the OT&E, on the other hand, is to evaluate the system's operational effectiveness and suitability through the use of realistic test scenarios, representative environments, and operations and maintenance personnel with the skill levels projected for eventual employment. If the development process has included

the operational emphasis that the system requirements intended, the task of OT&E is decidely easier. Unfortunately, sometimes operational characteristics are not translated effectively into *detailed system specifications, i.e.,* the desired operational intent may not be fully reflected.

OT&E brings to the system development cycle an independent view. The OT&E team, by observing system development activities throughout the development cycle, can provide independent advice on critical operational issues. This operational influence can best be exerted on the system development *early* in the cycle before significant resources have been committed to "metal-bending."

Genesis of Software OT&E.

The role of software weapon system evaluation was quickly recognized in the Air Force. The Air Force Test and Evaluation Center (AFTEC) history reflects[6]:

> There was also a major realignment of the AFTEC staff that occurred during the year, evolving from growing concern for effective operational test and evaluation of embedded computers in both airborne and fixed ground based systems.... The [ad hoc] group quickly agreed that AFTEC should not be deeply involved in computer systems per se, but rather that its interest lay more legitimately in systems, weapons, and command, control, and communications supported by computers.... by late spring, 1976, the need for organizational branches to manage AFTEC's growing involvement in the OT&E of computer software.... was recognized.

Thus began AFTEC's formal involvement in the operation test and evaluation of software. In a June 1978 report,[10] the GAO asserted:

> The Defense Department's plans and actions for improving software management do not sufficiently emphasize software test and evaluation. Mission performance, reliability, and maintainability are degraded because systems are produced and placed in operational use on the basis of insufficient software test and evaluation. Software needs to be thoroughly tested during development so that discrepancies are identified and corrected before the system is released to users.
>
> Software is an integral part of a weapon system; therefore, the same attention should be applied to plannng and performing software testing as is given to hardware. However, this is not often recognized by the developer, the tester, or the user, who are traditionally hardware oriented.

The report went on to say "although major systems depend heavily on software to perform critical mission functions, top management personnel have not fully considered software test results before making major decisions." Further, they added, "there are no standard OSD procedures for orderly software testing, and practices vary among projects even within the service."

The above criticisms were principally directed toward the DT&E for software although OT&E did not go uncriticized. The DT&E tests are dominated by module-oriented tests. As Littlewood points out[7] "it is an unfortunate fact of life that the integration phase usually reveals more failure modes than had been suspected during the time the individual modules were under test." Because of development schedules slips, etc., the system may come into OT&E with only limited integrated testing completed.

The software OT&E program at AFTEC was instituted in recognition of problems such as stated above. The GAO report recognized that software OT&E was not as effective as it should be. Improvement of methods and policies are a continuing concern and area of interest.

What Does Software OT&E Offer?

Goodenough says, in discussing computer program quality,

> Correctness is not necessary for a program to be useable and useful. Nor is correctness sufficient. A correct program may satisfy a narrowly drawn specification and yet not be suitable for operational use because in practice, inputs not satisfying the specification are presented to the program and the results of such incorrect usage are unacceptable to the user. If the program is correct with respect to an inadequate specification, its correctness is of little value.
>
> Consequently, although testing for correctness is the most common and best understood testing goal, correctness is by no means the only, important property of usable software - reliability, robustness, efficiency... are also of significant importance. But these properties are less commonly the focus of testing activities.[5]

As previously discussed, OT&E provides the bridge between DT&E and operational use. DT&E activities focus on specification compliance. As Goodenough points out, this is likely not an adequate test of operational usability. The focus of the software OT&E should be, then, not on compliance with specifications, but rather on the characteristics of software which are incompatible with actual operational conditions. The intent is to determine the acceptability of the system to the user, not only from a mission effectiveness point of view, but from a supportability point of view. In this context "the term 'acceptable' implies that the user must determine what he considers to be a failure; this usually depends on the effect of the particular behavior of the system in question on the user's operations, costs, etc."[13]

OT&E provides an opportunity to influence the operational characteristics of the software system. With access to program documentation, the OT&E team can independently assess the operational effect of specification (or other con-

tractual) changes. Apparent adverse effects can be used as a basis for test design. Software OT&E can also provide a basis for suggesting parameters/locations within software for redesign or modification.

CHARACTERISTICS OF SOFTWARE

In order to put evaluation of software into perspective, it is useful to understand some of the standard features of software, how software relates to hardware, and some of the desired features of software. This section discusses the above subjects with the intent of laying the foundations for determining the features of software evaluation during IOT&E.

ECS Characteristics.

Embedded computer systems (ECS) can be defined as follows:

> A system incorporated as an integral part of (dedicated and essential to the specific functional task for which the higher order system was designed) or required for direct support of (includes those functions such as specialized training, testing, or software support which are dedicated to the operation and maintenance of a system throughout its life-cycle) a major or less-than-major system.[14]

Note that this definition includes any automatic test equipment (ATE), support systems for code maintenance, training devices, etc. The salient characteristics of ECS are listed in table 1.

Hardware vs Software.

In ECS, which is a synergism of hardware and software, it is tempting to compare hardware and software characteristics and to apply the same measures for evaluation. This section will contrast hardware and software with respect to their life-cycles, failure mechanisms, and reliability.

Goodenough[5] contrasts the hardware and software life cycles as shown in table 2. His simplified comparison shows clearly that similar terms in the processes have strikingly different meanings. He summarizes the differences as:

a) Coding programs is not equivalent to manufacturing a product.
b) Maintenance refers to quite different processes.
c) Computer program development and test is conceptually similar to developing and testing a hardware prototype, but in software, the "prototype" is delivered to users.*

Thus, it is not enough, and perhaps highly misleading, to capitalize on similarities of terminology in order to evaluate software performance. It is imperative to understand the differences and ensure that evaluation criteria reflect those differences.

One important area of software/hardware difference is in the concept of "failure". Whereas hardware failures are almost always due to component deterioration (from age, temperature, humidity, etc.), software failures arise from design and/or implementation errors.** While software failures range from the relatively trivial to severe, "the occurrence of a system failure due to software failure is just as real to the user of the system as when due to a hardware failure."[4] While workarounds may exist to ease the impact of a failed function, that function is not available to the user.

It is also seen that hardware redundancy allows a system to be made as reliable as desired. This makes it convenient to specify reliability and to subsequently make cost/reliability tradeoffs in design. The same flexibility does not exist in software. For software, reliability is achieved by an adherence to good design principles combined with extensive testing. The cost tradeoffs for software reliability tend to occur during development, and the major factor is schedule adherence. Further, as we'll see in the next section, reliability for software is operationally difficult to define, and metrics for evaluation of reliability are not applicable to OT&E.

Adverse Software Characteristics.

This section will focus on adverse characteristics of software, the characteristics which tend to make software evaluation difficult during OT&E.

Software "Failure" Mechanisms.

As suggested earlier, software does not fail in the same sense as hardware. There are numerous reasons why software performance is classified a failure. The basic steps in developing a software system are:[5]

a) Defining user requirements.
b) Deciding what functions and major components a system must provide to meet those requirements.
c) Designing and specifying the intended behavior of understand software components.

* It has been recommended by Dodd[4] that software development strategies be modified to include development of an "operational prototype" of the software system. This prototype would aid in the development of user/customer/contractor agreement on requirements by engendering communciation on a common framework.

** A software failure can be viewed[9] as an operational malfunction-- a malfunction being ultimately any feature of operation that is unacceptable to the user and which occurs as a point event in time. When that malfunction is traced to a property of the software, the malfunction can be said to be software related, but we do not mean that the computer software fails in the same sense as the operational malfunction occurred.

d) Implementing the software components (coding).

Each of these software development activities is subject to error:

a) Construction errors. Failure of software components, as implemented, to satisfy their specifications.
b) Specification errors. Failure to accurately specify the intended behavior of a unit of software construction.
c) Functional design errors. Failure to establish an overall design able to meet identified requirements.
d) Requirement errors. Failure to identify user needs accurately, including failure to communicate these needs to the software designers.

MIL-STD-1679 classifies software problems as follows[11]:

a) Software trouble. The software does not operate according to support documentation and the documentation is right.
b) Documentation trouble. The software does not operate according to supporting documentation but the software operation is right.
c) Design trouble. The software operates according to supporting documentation but a design deficiency exists.
d) Logic trouble. The software has a logical error with no directly observable operational symptom but with the potential of creating trouble.

Software failures leave no signature. They take several forms:

a) The software does not respond to an input.
b) The software responds incorrectly to an input.
   1) Improperly timed response.
   2) Numerically wrong response.
   3) Response requiring more resources than are available.[15]

Further, while in concept the specification should track the operational requirements and the design should track the specification, divergences exist due to (principally) an inability to effectively communicate among the participants in the development cycle. Thus the operator's perception of a design which is in accordance with specifications may not be favorable, leading to a "software failure" report. In fact, this trait is a basic reason for operational testing. Correctness per specification does not necessarily imply operational utility. As noted previously "correctness is not necessary for a program to be usable and useful. Nor is correctness sufficient. If a program is correct with respect to an inadequate specification, its correctness is of little value."

Software Desired Features.

As previously quoted, "although testing for correctness is the most common and best understood testing goal, correctness is by no means the only important property of usable software - reliability, robustness, efficiency...are also of significant importance. But these properties are less commonly the focus of testing activities."

Since it is these "other" traits that operational testers are concerned with, this section will examine those features of software which are desireable to the user and maintainer. The following discussions are not intended to be exhaustive in the detailing of desired features, rather representative.

McCall, et al, divide software quality factors into three distinct stages of operation, as seen in figure 2. They assert that by taking a life-cycle view of software quality, appreciable savings in the total cost can be achieved. They maintain that "the major characteristics" that software systems have typically exhibited besides lack of reliability are the following:

a) High cost to maintain.
b) Lack of portability.
c) High sensitivity to changing requirements (inflexibility).
d) Lack of reusability.

Further details of the McCall model are shown in figure 3.[3] Boehm[2] similarly develops a software quality characteristics tree (figure 3). These two looks at software quality considerations provide a shopping list of characteristics which software should include.

Curtis[3] compares the ordering of characteristics by Boehm and McCall, both of whom have developed evaluation methodologies. He states...

> ...Both of these systems have been developed from an intuitive ordering of software characteristics. The higher level constructs in each system represent:
>
> a) The current behavior of the software.
> b) The ease of changing the software.
> c) The ease of converting or interfacing the system.

AFTEC, in its evaluation of software quality, divides the subject into two areas: effectiveness and suitability. The suitability area principally addresses maintainability as supported by the documentation and source listings (see figure 4). Table 3 is a comparision of characteristics associated by the above three sources for maintainability.

From figure 3 there are features other than maintainability which could be evaluated under suitability and which are desirable in computer programs to enhance their degree of legacy. These include reusability, portability, and interoperability. Definitions for these are found in figure 2. As McCall points out (reported by Curtis[3]) there are conflicts among the desired characteristics during development. For example, things done efficiently are not necessarily flexible, maintainable, etc. His analysis (using his model of figure 3) is shown in figure 5.

Figure 4 also addresses software quality features which are investigated during AFTEC

software effectiveness evaluation. Table 4 compares the features of the models with AFTEC evaluations.

Some features in table 4 which are not directly evaluated by AFTEC are indirectly evaluated as a result of close monitoring and evaluation of system problems. Others are not evaluated during IOT&E because the software is evaluated as a subsystem during total system operation.

Availability (or reliability) of the software during system operation is easily understood but difficult to define or quantify. Thus while these are highly desirable traits, direct evaluation of software availability is not made.

An area which merits consideration as an evaluation consideration is the reaction of the software to hardware features and recovery features (e.g., software/hardware interface timing, "graceful degradation," operator notification, etc.). Another potential area for software evaluation, especially in mission critical software, is software safety (i.e., can a software failure cause irrevocable damage to the equipment or operator and are adequate protections provided). A third area for consideration would be software security (e.g., design against unauthorized access or control of system).

McCall lists training as a feaure to be considered and it would appear that an AFTEC evaluation, as applicable, of the software training programs would be useful.

The other area of software quality features which is not addressed by McCall or Boehr is that of support resources. AFTEC does an evaluation of these resources and is pursuing means of providing a more consistent and useful evaluation. Among the features investigated typically are operation and support manning plans, the quality of the support system and its documentation, usability of the support system, and sufficiency of configuration management and quality assurance planning.

SOFTWARE OT&E AT AFTEC

As discussed in previous sections, the Air Force Test and Evaluation Center (AFTEC) has undertaken to evaluate software as part of the Operational Test and Evaluation (OT&E) of weapons systems. This section discusses some of the major problems associated with this evaluation, the current AFTEC approach to the evaluation, current initiatives to improve our capabilities, and, finally, some efforts required in the future.

Problems with Software OT&E.

Because of the role software naturally assumes in the integration of system elements, and because of the propensity to underestimate the complexity (hence schedule) of the software, software is typically on the critical path of any development and is a major cause of schedule slips. Among the consequences of this are:

a) The software is not 'mature'. That is, the software system has been inadequately tested, may not incorporate major functions, and will likely be subject to substantial change activity.
b) As tests uncover system hardware design deficiencies, the software tends to absorb the problems.
c) As the program schedule slips, the documentation integrity may be sacrificed.
d) Support facilities or software will remain in the contractor's facility for last minute change activity, hence not be available for evaluation.

A separate but related point is that the developed software under test is envisioned as that to be deployed. As a consequence, defects in design (from an operational view) will receive little attention if adequate workarounds are evident. Deficiencies in supportability will be overlooked (or required upgrades will be unfunded). The costs of upgrading software or documentation quality is prohibitive and, since these factors are not easily related to life-cycle-cost (of system support) the argument for upgrade is weak.

Aggravating these problems is a lack of test techniques or reporting methodology which adequately address software. The software must be tested as part of the system in an operational environment. This severly limits the strictly software considerations which could be evaluated. Further, once the testing is complete, reporting on software effectiveness is subjective and difficult to put into perspective with other system performance.

AFTEC Approach.

AFTEC has evolved a methodology for the evaluation of software during weapons systems OT&E. An evaluation tree was shown in figure 4, and was discussed in previous sections. Briefly, the evaluation focuses on two aspects of the software: the software's effectiveness in the system's accomplishment of the operational mission; and the extent to which the software system supports the mission and is maintainable (suitability). Arner, etal[1] provide a more detailed discussion of the AFTEC approach.

Suitability.

The evaluation of this attribute is focused on the maintainability of the software system, the adequacy and effectiveness of the computer support resources, and the effectiveness of the support software (e.g., off-line diagnostics). The evaluation of maintainability is supported by standardized questionnaires which are filled out by evaluators experienced in maintaining software. Random samples of the source listings (about 10 percent) and the documentation are carefully reviewed and an analysis provided via machine readable answer sheets. Three to five evaluators are used for each module evaluted to provide a statistical basis for analysis. The statistical analysis is provided by computer programs at HQ AFTEC. The areas specifically assessed are

modularity, descriptiveness, consistency, simplicity, expandability, and instrumentation.

The computer support resources are evaluated by a variety of methods depending on the status of planning or acquisition. The evaluation is supported by personnel from the Air Logistics Center or from the using command familiar with such facilities. The objective is to determine whether extant planning will provide sufficient and adequate support capability for making changes to the mission software.

Effectiveness.

The evaluation of this attribute is less straight forward. From figure 4, the software operator-machine interface evaluation focuses on the effectiveness of the software to communicate system status and function to the operator(s). That evaluation is conducted via a standard questionnaire processed as for the maintainability questionnaire. In this case, system operators are the evaluators. The areas assessed in this evaluation are assurability, controllability, workload reasonability, descriptiveness, consistency, and simplicity.

However, the operational effectiveness of the system software is difficult to evaluate. There are no established methodologies or metrics to support the evaluation. The standard means of assessing this feature is to track system errors that are allocated to software and from the frequency and severity of software problems, make a judgement, albeit subjective, of the software readiness for operations.

A computer logic and performance monitor known as the Event Trace Monitor (ETM), developed by AFTEC, can be used to provide accurate estimates of timing margins of computer programs under operational conditions. The ETM role is still being developed for OT&E.

Current AFTEC Activities.

AFTEC continues to try to improve their ability to more effectively evaluate weapons system software. Two initiatives currently being pursued are development of a standard questionnaire type approach to evaluation of computer support resources, and tasking of Independent Verification and Validation (IV&V) contractors.

The computer support resources evaluation methodology is currently being developed under contract and we anticipate a demonstration of the capability by the end of 1981. This methodology will significantly enhance our capabilities to evaluate support resources by allowing us to tailor a questionnaire approach to the circumstances, and thus ensure that the appropriate questions are asked. A quantitative evaluation result will then be derived.

Our increased involvement with the IV&V contractor, when available, implies that tasks will be developed which will result in enhanced planning information being available. IV&V contractors will, in concept, provide their analysis to AFTEC, to include an analysis of potential operational weaknesses and candidate test scenarios to determine the extent of the weakness. AFTEC has initiated trial efforts on two programs to determine the soundness of this approach.

What Needs to be Done.

As mentioned previously, one weakness of the AFTEC approach to evaluation of software effectiveness is lack of a quantitative methodology or of useful metrics. This is an area which merits more thought. Questions to be asked include:

a) Does a metric exist which describes software effectiveness (e.g., reliability, maturity, availability) and would be useful to a decision maker?
b) Can data derived during OT&E be used to satisfactorily estimate the metric?
c) Can other data (e.g., from development test and evaluation) be used to estimate the metric?
d) Can a standardized methodology be developed to evaluate the operational effectiveness of software?

SUMMARY

AFTEC has been evaluating software during weapons systems OT&E since 1976. A series of methodologies for this evaluation have evolved that provide a reasonable overall assessment of the software, but there are areas for continued improvement.

REFERENCES

1. Arner, H.L., F.B. Fisk, C.E. Martin, "AFTEC Software Operational Test and Evaluation Methodology for Electronic Warfare Systems," Proceedings of the Twenty Fifth Annual Joint Electronic Warfare Conference, Vol 2, p. 521.

2. Boehm, B. W., J. R. Brown, and M. Lipon. "Quantitative Evaluation of Software Quality," Proceedings and International Conference on Software Engineering, 1976.

3. Curtis, Bill, "Measurement and Experimentation in Software Engineering," IEEE Proceedings, September 1980, p. 1144.

4. Dodd, W. P. "Prototype Program," Computer, February 1980, p. 81.

5. Goodenough, J. B. and Clement McGowan "Software Quality Assurance: Testing and Validation," IEEE Proceedings, September 1980, p. 1093.

6. History of the Air Force Test and Evaluation Center, 1 Jan 1976-31 Dec 1976, Vol I.

7. Littlewood, Bev. "Software Reliability Model for Modular Program Structure," IEEE Transactions on Reliability, Aug 1979, p.241.

8. Lipow, M. "On Software Reliability. A Preface by the Guest Editor." IEEE Transactions on Reliability, August 1979, p. 178.

9. Littlewood, B. "How to Measure Software Reliability and How Not To." IEEE Transactions of Reliability, June 1979, p. 103-110.

10. Managing Weapon System Software: Progress and Problems, GAO, Report to the Congress, July 10, 1978.

11. Mil-Std 1679 (Navy), 2 December 1978.

12. Murch, Walter G. Embedded Computer Systems-Implications for Fixed Price Procurement, Air Command and Staff College, Maxwell AFB, AL, Research Report 1730-80, May 1980.

13. Musa, John D. "The Measurement and Management of Software Reliability," IEEE Proceedings, Sep 80, p.1131.

14. Riski, William A. "Software Maintainability Characteristics;" Conference Briefing, Proceedings, Software Operational Support in the 1980s, December 1980.

15. Software Testing, Test Operations Procedure TOP 1-1-056. U.S. Army Test And Evaluation Command, Aberdeen Proving Ground, MD, November 1977, (AD A046962).

16. Walters, A. F. and Jim McCall. "Software Quality Metrics for Life Cycle Cost Reduction; IEEE Transactions on Reliability, August 1979, p. 212.

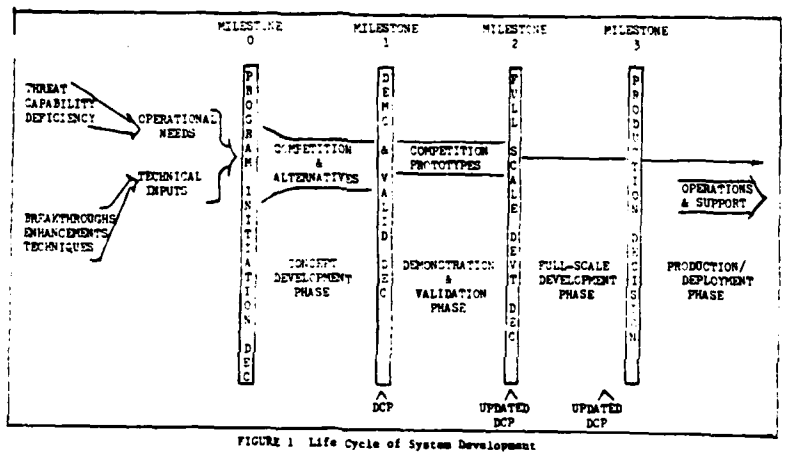

FIGURE 1 Life Cycle of System Development

INITIAL PRODUCT OPERATION	CORRECTNESS	Extent to which a program satisfies its specifications and fulfills the user's objectives.
	RELIABILITY	Extent to which a program can be expected to perform its intended function with required precision.
	EFFICIENCY	The amount of computing resources and code required by a program to perform a function.
	INTEGRITY	Extent to which access to software or data by unauthorized persons can be controlled.
	USABILITY	Effort required to learn, operate, prepare input, and interpret output of a program.
PRODUCT REVISION	MAINTAINABILITY	Effort required to locate and fix an error in an operational program.
	TESTABILITY	Effort required to test a program to ensure it performs its intended function.
	FLEXIBILITY	Effort required to modify an operational program.
PRODUCT TRANSITION	PORTABILITY	Effort required to transfer a program from one hardware configuration and/or software system environment to another.
	REUSABILITY	Extent to which a program can be used in other applications – related to the packaging and scope of the functions that programs perform.
	INTEROPERABILITY	Effort required to couple one system with another.

FIGURE 2 Software Quality Factors

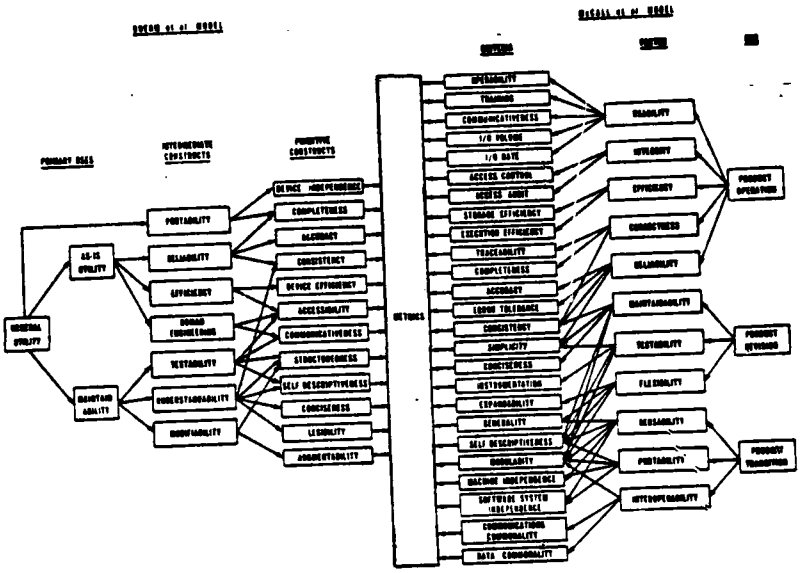

FIGURE 3 The McCall and Boehm Models

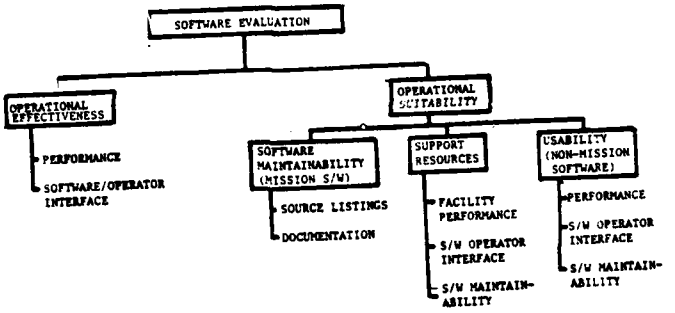

FIGURE 4 AN AFTEC SOFTWARE EVALUATION

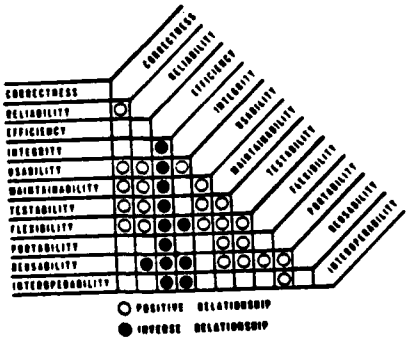

FIGURE 5 Software Characteristic Conflicts

EMBEDDED COMPUTER SYSTEM (ECS)

An ECS is integral to an electronic or electromechanical system (for example, combat weapon system, tactical system, aircraft, ship, missile, spacecraft, command, control and communication systems) from a design, procurement and operations viewpoint.

Key Attributes

- Developed, acquired and operated under decentralized management (DOD Directives 5000.1 and 5000.2).
- Physically incorporated into a larger system whose function is not data processing.
- Integral to, or supportive of, a larger system from a design, procurement and operations viewpoint.
- Outputs include information, control signals and computer data.

Detailed Characteristics

- Simultaneous hardware and software development.
- Transportable/deployable (generally militarized).
- Special purpose or one-of-a-kind.
- Programs machine-dependent.
- Designed to fit into larger, non-automatic data processing (ADP) system.
- Tailored programming languages.
- Specialized computer equipment.
- Development/acquisition/support as a configuration item.
- Need high reliability software.
- Extensive/expensive test programs.

TABLE 1 Embedded Computer System Characteristics

	HARDWARE	SOFTWARE
DEVELOPMENT	Determine User Requirements	Determine User Requirements
	Develop Product Concept (Functional Design)	Develop Product Concept (Functional Design)
	Specify Component Design (Detailed Design)	Specify Component Design (Detailed Design)
	Build and Test Prototype	*Implement and Test Programs
	Develop Manufacturing Techniques	--
INSTALLATION	Manufacture Product	*Copy Programs
	Make Product Available to User	Make Program Available to User
MAINTENANCE/ IMPROVEMENT	Maintenance (Correct Component Failures)	* --
	Recall Product to Correct Design Flaws	*Maintenance (Correct Implementation and Design Errors)
	Enhance Product	Maintenance (Provide Enhanced Capabilities; Adapted to Changed User Environment)
PHASE-OUT	Unit is Unusable and Unrepairable (replace with new unit)	--
	Product is Obsolete	Product is Obsolete

*marks important differences in terminology

TABLE 2 Corresponding Steps of S/W and H/W Product Life Cycles

Boehm	McCall	AFTEC
Consistency	Consistency	Consistency
Accessibility		
Communicativeness		
Structuredness	Modularity	Modularity
Self-Descriptiveness	Self-descriptiveness	Descriptiveness
Conciseness	Conciseness	
Legibility	Simplicity	Simplicity
Augmentability	Expandibility	Expandibility
	Generality	
	Instrumentation	Instrumentation

TABLE 3 Comparison of Software Features (Maintainability)

Boehm	McCall	AFTEC
Completeness	Completeness	
Accuracy	Accuracy	
Consistency	Consistency	
Device efficiency	storage, execution } efficiency	efficiency { timing, memory
	I/O volume	
	I/O rate	
Accessibility		
Communications	Communicativeness	Communicativeness
	access control/audit	(S/W operator
	traceability	interface)
	error tolerance	
	simplicity	
	training	
	operability	(S/W operator interface)

TABLE 4 Comparison of Software Features (Effectiveness)